

**IMPLEMENTASI THREE.JS DAN WEBXR DALAM PENGEMBANGAN
APLIKASI *VIRTUAL REALITY* (VR) UNTUK PENGENALAN
PERANGKAT KERAS KOMPUTER**

TUGAS AKHIR



FALACHUL AKHADIHIMA IBRAHAICAL
NIM: 311910026

UNIVERSITAS
MA CHUNG

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI DAN DESAIN
UNIVERSITAS MA CHUNG
MALANG
2026**

LEMBAR PENGESAHAN
TUGAS AKHIR

**IMPLEMENTASI THREE.JS DAN WEBXR DALAM PENGEMBANGAN
APLIKASI VIRTUAL REALITY (VR) UNTUK PENGENALAN
PERANGKAT KERAS KOMPUTER**

Oleh:

**Falachul Akhadihima Ibrahaical
NIM. 311910026**

dari:

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI DAN DESAIN
UNIVERSITAS MA CHUNG**

Telah dinyatakan lulus dalam melaksanakan Tugas Akhir sebagai syarat kelulusan
dan berhak mendapatkan gelar Sarjana Teknik S.Kom.

Dosen Pembimbing I,



Paulus Lucky Tirma Irawan, .Kom., MT.
NIP. 20100005

Dosen Pembimbing II,



Bita Parga Zen, S.Kom., M.Han.
NIP. 20240017

Dekan Fakultas Teknologi dan Desain,



Prof. Dr. Eng. Romy Budhi Widodo
NIP. 2007003

PERNYATAAN KEASLIAN SKRIPSI

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Skripsi saya dengan “Implementasi Three.Js Dan WebXR dalam Pengembangan Aplikasi *Virtual Reality* (VR) untuk Pengenalan Perangkat Keras Komputer” adalah benar benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Malang, 7 Januari 2026

Falachul Akhadihima Ibrahaical
NIM. 311910026

UNIVERSITAS
MA CHUNG

**IMPLEMENTASI THREE.JS DAN WEBXR DALAM PENGEMBANGAN
APLIKASI *VIRTUAL REALITY* (VR) UNTUK PENGENALAN
PERANGKAT KERAS KOMPUTER**

Falachul Akhadihima Ibrahaical, Paulus Lucky Tirma Irawan,

Bita Parga Zen

Universitas Ma Chung

Abstrak

Pemahaman perangkat keras komputer merupakan aspek penting dalam pendidikan teknologi, namun penyampaian materi masih didominasi media 2D yang kurang interaktif. Teknologi *Virtual Reality* (VR) menawarkan pembelajaran imersif, tetapi platform seperti Unity sering menghasilkan ukuran aplikasi besar dan membutuhkan perangkat keras tinggi. Penelitian ini mengembangkan aplikasi VR berbasis web yang ringan untuk pengenalan perangkat keras komputer menggunakan Three.js dan WebXR. Metode pengembangan yang digunakan adalah *Multimedia Development Life Cycle* (MDLC) dengan optimasi aset 3D melalui Blender dan glTF-Transform. Hasil penelitian menunjukkan peningkatan performa signifikan, meliputi penurunan ukuran *file* sebesar 84,97%, pengurangan VRAM 80,40%, dan penurunan *draw calls* 71,21%. Pengujian performa menunjukkan *frame rate* stabil pada 60 FPS di mode *desktop* dan 72–90 FPS di VR. Pengujian fungsional dan usabilitas menghasilkan skor SUS 78,83, menandakan aplikasi efektif dan mudah diakses.

Kata Kunci: *Virtual Reality*, WebXR, Three.js, Optimasi Aset, Perangkat Keras Komputer

**IMPLEMENTATION OF THREE.JS AND WEBXR IN THE DEVELOPMENT
OF A VIRTUAL REALITY (VR) APPLICATION FOR INTRODUCING
COMPUTER HARDWARE**

Falachul Akhadihima Ibrahaical, Paulus Lucky Tirma Irawan,

Bita Parga Zen

Universitas Ma Chung

Abstract

Understanding computer hardware is essential in technology education, yet learning materials often rely on static 2D media. Virtual Reality (VR) offers immersive learning, but platforms such as Unity typically produce large application sizes and require high-spec hardware, limiting accessibility. This study develops a lightweight, web-based VR application for introducing computer hardware using Three.js and WebXR. The development process follows the Multimedia Development Life Cycle (MDLC) with a focus on 3D asset optimization using Blender and glTF-Transform. The results show significant performance improvements, including an 84.97% reduction in file size, an 80.40% decrease in VRAM usage, and a 71.21% reduction in draw calls. Performance testing demonstrates stable frame rates of 60 FPS on desktop mode and 72–90 FPS in VR. Functional and usability evaluations indicate that all features operate correctly, and the System Usability Scale (SUS) score of 78.83 confirms that the application is effective, accessible, and well-received by users.

Keywords: Virtual Reality, WebXR, Three.js, Asset Optimization, Computer Hardware

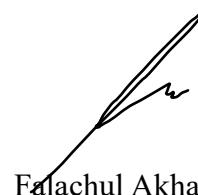
KATA PENGANTAR

Puji syukur ke hadirat Tuhan Yang Maha Esa atas segala rahmat, hidayah, dan karunia-Nya, sehingga penulis dapat menyelesaikan Laporan Tugas Akhir yang berjudul "Implementasi Three.js dan WebXR dalam Pengembangan Aplikasi *Virtual Reality* (VR) untuk Pengenalan Perangkat Keras Komputer". Laporan Tugas Akhir ini disusun sebagai salah satu syarat untuk menyelesaikan pendidikan Strata Satu (S1) pada Program Studi Teknik Informatika, Fakultas Teknologi dan Desain, Universitas Ma Chung. Penelitian ini bertujuan untuk mengembangkan media pembelajaran interaktif berbasis web yang imersif guna membantu pemahaman mengenai perangkat keras komputer. Penyusunan laporan ini tidak lepas dari bantuan, bimbingan, dan dukungan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

- 1) Bapak Paulus Lucky Tirma Irawan, S.Kom., MT., selaku dosen pembimbing pertama pelaksanaan Tugas Akhir.
- 2) Bapak Bita Parga Zen, S.Kom., M.Han., selaku dosen pembimbing kedua pelaksanaan Tugas Akhir.
- 3) Bapak Hendry Setiawan, ST., M.Kom., selaku dosen penguji pelaksanaan Tugas Akhir.
- 4) Segenap Dosen dan Staf Program Studi Teknik Informatika Universitas Ma Chung atas ilmu dan pelayanan yang diberikan selama masa perkuliahan.

Penulis menyadari bahwa laporan Tugas Akhir ini masih jauh dari kesempurnaan, baik dari segi penyusunan, bahasa, maupun materi. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang membangun dari pembaca guna perbaikan di masa mendatang.

Malang, 7 Januari 2026



Falachul Akhadihima Ibrahaical

DAFTAR ISI

DAFTAR ISI.....	i
DAFTAR TABEL.....	iii
DAFTAR GAMBAR	v
Bab I Pendahuluan	1
1.1 Latar Belakang	1
1.2 Identifikasi Masalah.....	3
1.3 Batasan Masalah.....	4
1.4 Perumusan Masalah	4
1.5 Tujuan Penelitian	4
1.6 Manfaat Penelitian	5
1.7 Luaran Penelitian	5
1.8 Sistematika Penulisan	6
Bab II Tinjauan Pustaka	7
2.1 Three.js.....	7
2.2 WebXR.....	8
2.3 <i>Virtual Reality (VR)</i>	9
2.4 Perangkat Keras Komputer (<i>Hardware</i>)	12
2.5 HTML (<i>Hypertext Markup Language</i>)	14
2.6 CSS (<i>Cascading Style Sheets</i>).....	15
2.7 JavaScript	15
2.8 WebGL.....	16
2.9 Pemodelan 3D (<i>3D Modeling</i>)	18
2.10 Blender	19
2.11 glTF-Transform.....	19
2.12 glTF-Report.....	20
2.13 GPT-SOVITS	20
2.14 Github.....	23
2.15 Cloudflare Pages	23
2.16 WebXR API Emulator	24
2.17 Pengujian <i>Black Box</i>	24
2.18 <i>System Usability Scale (SUS)</i>	25

2.19	Penelitian Terdahulu	26
Bab III	Analisis dan Perancangan Sistem	30
3.1	Alur Penelitian	30
3.2	<i>Concept</i> (Konsep).....	31
3.3	<i>Design</i> (Perancangan)	33
3.4	<i>Material Collection</i> (Pengumpulan Materi).....	44
3.5	<i>Assembly</i> (Pembuatan)	47
3.6	<i>Testing</i> (Pengujian)	52
3.7	<i>Distribution</i> (Distribusi).....	57
Bab IV	Hasil dan Pembahasan	58
4.1	Hasil Konsep dan Perancangan.....	58
4.2	Hasil Pengumpulan Materi.....	69
4.3	Hasil Pembuatan.....	72
4.4	Hasil Pengujian	80
4.5	Hasil Distribusi	85
Bab V	Simpulan dan Saran	87
5.1	Simpulan	87
5.2	Saran.....	88
DAFTAR PUSTAKA	89
Lampiran A	Naskah Narasi Audio.....	93
Lampiran B	Kode Utama Aplikasi.....	98
Lampiran C	Kuesioner di Google Form.....	107
Lampiran D	Hasil Kuesioner	111

DAFTAR TABEL

Tabel 2. 1 Deskripsi Struktur Dasar Tag HTML	14
Tabel 2. 2 Jenis Penerapan CSS pada Dokumen HTML	15
Tabel 2. 3 Jenis Penerapan JavaScript pada Dokumen HTML.....	16
Tabel 2. 4 Perintah CLI glTF-Transform dan Fungsinya	20
Tabel 2. 5 Fitur Utama GitHub	23
Tabel 2. 6 Instrumen Pernyataan <i>System Usability Scale</i> (SUS)	25
Tabel 2. 7 Analisis <i>Keterbatasan</i> Studi Terdahulu	27
Tabel 3. 1 Deskripsi <i>Use Case</i> Aplikasi	35
Tabel 3. 2 <i>Storyboard</i> Aplikasi	40
Tabel 3. 3 <i>Storyboard</i> Aplikasi (Lanjutan)	41
Tabel 3. 4 <i>Storyboard</i> Aplikasi (Lanjutan)	42
Tabel 3. 5 <i>Storyboard</i> Aplikasi (Lanjutan)	43
Tabel 3. 6 Kriteria Pemilihan Model 3D.....	45
Tabel 3. 7 Rancangan Hasil Pengumpulan Model 3D	45
Tabel 3. 8 Rancangan Hasil Pengumpulan Model 3D (Lanjutan)	46
Tabel 3. 9 Rancangan Ringkasan Penurunan <i>Vertex</i> pada Model 3D.....	49
Tabel 3. 10 Rancangan Ringkasan Penurunan VRAM pada Model 3D	50
Tabel 3. 11 Rancangan Ringkasan Penurunan <i>Draw Call</i> pada Model 3D	50
Tabel 3. 12 Rancangan Ringkasan Penurunan Ukuran <i>Disk</i> pada Model 3D.....	51
Tabel 3. 13 Rancangan Skenario Pengujian Fungsionalitas <i>Black Box</i>	53
Tabel 3. 14 Rancangan Skenario Pengujian Fungsionalitas <i>Black Box</i> (Lanjutan)	54
Tabel 3. 15 Spesifikasi Perangkat Pengujian	54
Tabel 3. 16 Rancangan Uji Performa Pengukuran Waktu Muat Model 3D	55
Tabel 3. 17 Rancangan Uji Performa Pengukuran FPS	56
Tabel 3. 18 Format Kuesioner SUS	57
Tabel 3. 19 Rincian Distribusi Aplikasi.....	57
Tabel 4. 1 Hasil Pengumpulan Model 3D.....	70
Tabel 4. 2 Hasil Pengumpulan Model 3D (Lanjutan)	71
Tabel 4. 3 Ringkasan Penurunan <i>Vertex</i> pada Model 3D	72
Tabel 4. 4 Ringkasan Penurunan VRAM pada Model 3D	73

Tabel 4. 5 Ringkasan Penurunan <i>Draw Call</i> pada Model 3D.....	74
Tabel 4. 6 Ringkasan Penurunan Ukuran <i>Disk</i> pada Model 3D	74
Tabel 4. 7 Hasil Pengujian Fungsional <i>Black Box</i>	80
Tabel 4. 8 Hasil Pengujian Fungsional <i>Black Box</i> (Lanjutan)	81
Tabel 4. 9 Hasil Uji Performa Pengukuran Waktu Muat Model 3D.....	82
Tabel 4. 10 Hasil Uji Performa Pengukuran FPS	82
Tabel 4. 11 Hasil Uji Performa Pengukuran FPS (Lanjutan)	83
Tabel 4. 12 Hasil Pengujian Usabilitas (SUS)	84

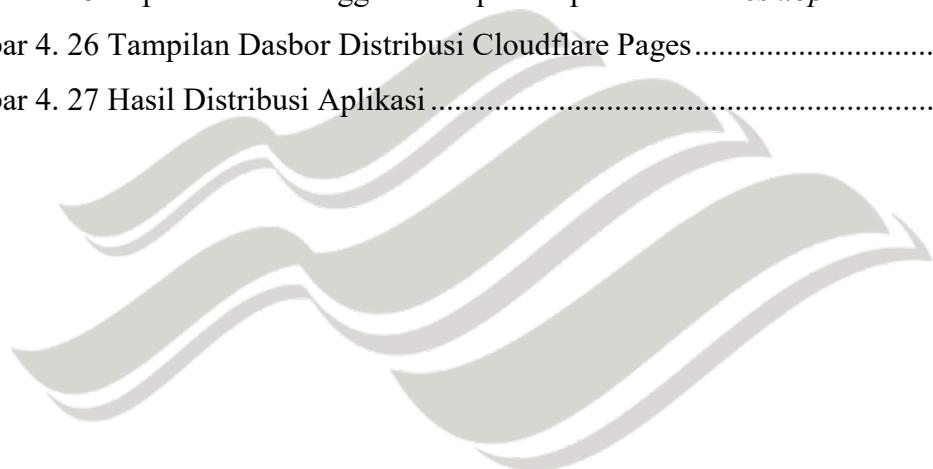


UNIVERSITAS
MA CHUNG

DAFTAR GAMBAR

Gambar 2. 1 Contoh Kode Struktur Dasar HTML.....	14
Gambar 2. 2 Struktur Dasar Objek 3D	18
Gambar 2. 3 Arsitektur Model GPT-SoVITS (Srivastava, 2025).....	21
Gambar 2. 4 Contoh Alur Penelitian dengan MDLC (Anggoro dkk., 2024).....	28
Gambar 3. 1 Alur Penelitian.....	30
Gambar 3. 2 Diagram <i>Flowchart</i>	34
Gambar 3. 3 Diagram <i>Use Case</i> Aplikasi	35
Gambar 3. 4 Diagram <i>Activity</i> Mengelola Sesi	36
Gambar 3. 5 Diagram <i>Activity</i> Mulai Sesi Belajar	37
Gambar 3. 6 Diagram <i>Activity</i> Mengerjakan Tes Akhir	38
Gambar 3. 7 Diagram <i>Activity</i> Melihat Laporan Belajar	39
Gambar 3. 8 Proses Pengumpulan Materi.....	44
Gambar 3. 9 Proses Optimasi Model 3D	48
Gambar 3. 10 Arsitektur Implementasi Aplikasi Berbasis Three.js dan WebXR.	51
Gambar 4. 1 Tampilan Layar Pemuatan	58
Gambar 4. 2 Tampilan Halaman Sampul.....	59
Gambar 4. 3 Tampilan Halaman Input Nama	60
Gambar 4. 4 Tampilan Halaman Tutorial	60
Gambar 4. 5 Tampilan Halaman Pilih Mode	61
Gambar 4. 6 Tampilan Halaman Menu Utama	62
Gambar 4. 7 Halaman Pilih Materi	63
Gambar 4. 8 Halaman Pembelajaran Materi	63
Gambar 4. 9 Halaman Mini Kuis	64
Gambar 4. 10 Hasil Halaman Umpang Balik Mini Kuis	65
Gambar 4. 11 Hasil Halaman Sesi Belajar Selesai	65
Gambar 4. 12 Halaman Pilih Materi (Akses Tes Akhir).....	66
Gambar 4. 13 Hasil Halaman Soal Final <i>Test</i>	67
Gambar 4. 14 Hasil Halaman Umpang Balik Final <i>Test</i>	67
Gambar 4. 15 Hasil Halaman Laporan Akhir	68
Gambar 4. 16 Hasil Halaman Sesi Belajar Berakhir	69

Gambar 4. 17 Tampilan Repositori Proyek di Github	75
Gambar 4. 18 Tampilan Dasbor Cloudflare Pages.....	76
Gambar 4. 19 Inisialisasi komponen inti Three.js.....	76
Gambar 4. 20 Inisialisasi <i>Raycaster</i>	77
Gambar 4. 21 Konfigurasi sesi <i>Virtual Reality</i> (VR).....	78
Gambar 4. 22 Tampilan Hasil <i>Render Mode Desktop</i>	79
Gambar 4. 23 Tampilan Hasil <i>Render Mode VR</i>	79
Gambar 4. 24 Mekanisme <i>Render</i> dan <i>Update Frame</i>	79
Gambar 4. 25 Implementasi Penggunaan Aplikasi pada Mode <i>Desktop</i> dan VR.	83
Gambar 4. 26 Tampilan Dasbor Distribusi Cloudflare Pages.....	85
Gambar 4. 27 Hasil Distribusi Aplikasi	86



UNIVERSITAS
MA CHUNG

Bab I

Pendahuluan

1.1 Latar Belakang

Komputer merupakan komponen penting dalam kehidupan sehari-hari yang kinerjanya dipengaruhi secara signifikan oleh perangkat keras. Perangkat keras menjalankan serangkaian fungsi fundamental, mulai dari eksekusi instruksi, pemrosesan informasi, hingga penyimpanan data, sehingga menjadi fondasi penting bagi sistem komputer dan seluruh aplikasi di dalamnya (Abuhamra, 2023; Erkamim, 2023). Meskipun demikian, pemahaman mengenai perangkat keras, khususnya di kalangan pelajar atau mahasiswa, seringkali terbatas. Kondisi ini disebabkan oleh materi pembelajaran yang pada umumnya disampaikan secara konvensional melalui buku teks, yang hanya menyajikan informasi dalam format dua dimensi (2D) dan kurang interaktif (Repiliya dan Mair, 2021; Purwoko dan Zen, 2023; Landa dkk., 2025; Buchori dkk., 2025).

Untuk mengatasi keterbatasan tersebut, diperlukan suatu pendekatan pembelajaran yang lebih interaktif dan imersif. Salah satu teknologi yang menawarkan solusi adalah *Virtual Reality* (VR), yang mampu menciptakan lingkungan simulasi tiga dimensi (3D) sehingga pengguna dapat berinteraksi secara langsung dengan objek virtual (De Luna, 2022; Budharti dkk., 2022). VR telah terbukti efektif dalam meningkatkan pemahaman konseptual, terutama untuk materi yang memerlukan visualisasi spasial kompleks seperti perangkat keras komputer (Azmi dkk., 2024; Hardiana dkk., 2025). Dengan memfasilitasi eksplorasi dan interaksi terhadap model 3D, VR dapat memberikan pengalaman pembelajaran yang lebih mendalam dan mudah dipahami dibandingkan metode konvensional.

Dalam pengembangan konten 3D dan VR, *game engine* seperti Unity seringkali menjadi pilihan utama karena didukung oleh ekosistem yang matang, antarmuka pengguna yang intuitif, serta kapabilitas lintas platform yang luas. Sebagai platform yang populer, Unity menyediakan alur kerja terintegrasi mulai dari tahap desain, pengembangan, hingga penerapan (*deployment*). Oleh karena itu,

platform ini sering diposisikan sebagai solusi komprehensif untuk menciptakan pengalaman imersif yang kompleks, sehingga banyak aplikasi edukasi dan simulasi dikembangkan menggunakananya sebagai standar industri (Kurniasari dkk., 2023; Lee, 2024; Fitriyanto dkk., 2024; Irawan dan Rahmadianto, 2025).

Akan tetapi, implementasi Unity untuk aplikasi edukasi berbasis web yang menuntut aksesibilitas tinggi dan efisiensi biaya justru menghadapi serangkaian tantangan signifikan. Dari aspek finansial, model lisensi Unity dapat menjadi kendala bagi institusi pendidikan maupun pengembang Non Komersial, sebab fitur-fitur penting untuk optimasi dan proses *deployment* hanya tersedia pada versi berbayar. Dari aspek teknis, saat aplikasi diekspor ke platform web (WebGL), ukuran *file* yang dihasilkan cenderung besar dan menuntut konsumsi memori yang tinggi. Hal ini mengakibatkan waktu muat yang lebih lama dan performa yang kurang stabil, terutama pada perangkat dengan spesifikasi standar. Di samping itu, ketergantungan pada *runtime* eksternal bertentangan dengan prinsip aksesibilitas instan yang menjadi keunggulan teknologi web (Hussain et al., 2020). Berbagai keterbatasan tersebut mengindikasikan perlunya eksplorasi terhadap arsitektur alternatif yang lebih ringan dan terbuka.

Sebagai respons terhadap tantangan-tantangan tersebut, berkembanglah pendekatan melalui teknologi web modern seperti WebXR, yang memungkinkan pengalaman 3D dan VR dapat diakses langsung melalui peramban tanpa memerlukan instalasi aplikasi atau perangkat keras khusus (Fachri dan Darmawan, 2022; Wibowo, 2022; Matahari, 2022; Mouttalib et al., 2023). Teknologi ini membuka peluang untuk menghadirkan konten interaktif yang mudah dijangkau dalam lingkup pendidikan. Dalam implementasinya, Three.js menjadi salah satu pustaka JavaScript yang fundamental untuk membangun grafis 3D di web. Pustaka ini menyederhanakan proses pengembangan dengan menyediakan antarmuka yang lebih mudah digunakan di atas WebGL, akibatnya pengelolaan objek, pencahayaan, tekstur, dan animasi dapat dilakukan secara lebih efisien (Astianingrum dkk., 2025; Straccia, 2024).

Namun, implementasi Three.js juga memiliki tantangan teknis tersendiri, terutama terkait optimalisasi performa *rendering* dan model 3D yang kompleks

(Putra dkk., 2022; Nuswantoro dkk., 2024). Sebab itu, menjaga kinerja aplikasi tetap optimal menjadi prioritas untuk menghadirkan pengalaman belajar yang imersif serta meminimalkan potensi gangguan fisik seperti *motion sickness* (Fanini et al., 2021; Wibowo, 2022). Selain itu, proses pengembangan perlu merancang mekanisme kontrol yang konsisten untuk berbagai perangkat masukan, seperti *mouse* pada *desktop* dan *controller* pada *headset VR* (Rantamaa et al., 2023; Nur dkk., 2024).

Berdasarkan serangkaian tantangan tersebut, penelitian ini berfokus pada pengembangan aplikasi pembelajaran perangkat keras komputer berbasis VR menggunakan Three.js dan WebXR yang dirancang melalui metode *Multimedia Development Life Cycle* (MDLC). Aplikasi ini diharapkan dapat menjadi media pembelajaran interaktif yang imersif dan mudah diakses melalui web, dengan performa terukur untuk meningkatkan pemahaman mahasiswa mengenai perangkat keras komputer.

1.2 Identifikasi Masalah

Berdasarkan uraian pada latar belakang, permasalahan utama dalam pembelajaran perangkat keras komputer saat ini adalah ketergantungan pada media konvensional, seperti buku teks, yang hanya menyajikan informasi dalam format dua dimensi dan bersifat statis. Keterbatasan visualisasi ini menyebabkan mahasiswa kurang mendapatkan pengalaman interaktif, sehingga mereka kesulitan untuk memahami struktur ruang dan bentuk fisik komponen komputer secara mendalam.

Selain itu, dari sisi teknis pengembangan, penggunaan platform populer seperti Unity memiliki hambatan signifikan terkait lisensi berbayar dan efisiensi sumber daya. Aplikasi VR yang dihasilkan cenderung memiliki ukuran *file* yang besar dan konsumsi memori tinggi yang membebani perangkat standar. Aksesibilitas juga menjadi kendala utama karena aplikasi native mengharuskan pengguna melakukan instalasi tambahan, sehingga media pembelajaran menjadi tidak praktis dan sulit diakses secara instan.

1.3 Batasan Masalah

Untuk menjaga fokus dan kelancaran dalam penelitian ini, batasan masalah yang ditetapkan adalah sebagai berikut:

- 1) Aplikasi dirancang untuk berjalan pada *browser* web modern yang mendukung WebXR, dengan target perangkat *desktop* dan *headset VR*.
- 2) Materi perangkat keras komputer dibatasi pada komponen-komponen utama, yaitu monitor, *keyboard*, *motherboard*, *mouse*, CPU, memori, GPU, kartu jaringan, *storage*, printer, dan flashdisk.
- 3) Model 3D yang digunakan dalam aplikasi diperoleh dari platform daring seperti Sketchfab dan sumber sejenisnya.
- 4) Interaksi pengguna dalam aplikasi dibatasi pada fitur dasar, yaitu kontrol kamera, navigasi menu, penyajian informasi tekstual dan audio, serta mekanisme evaluasi berupa mini kuis dan tes akhir.

1.4 Perumusan Masalah

Berdasarkan identifikasi dan batasan masalah yang telah dijelaskan, rumusan masalah dalam penelitian ini adalah:

- 1) Bagaimana arsitektur aplikasi berbasis Three.js dan WebXR dapat dirancang untuk menghasilkan kinerja optimal pada mode *desktop* maupun perangkat VR?
- 2) Bagaimana optimasi model 3D dapat dilakukan untuk menjaga performa aplikasi tetap stabil?
- 3) Bagaimana merancang mekanisme interaksi yang konsisten agar perintah dari *mouse* di *desktop* dan *controller* di VR diterjemahkan menjadi aksi yang seragam dan intuitif bagi pengguna?

1.5 Tujuan Penelitian

Penelitian ini bertujuan untuk mengembangkan dan mengevaluasi implementasi Three.js dan WebXR sebagai solusi alternatif aplikasi *Virtual Reality* (VR) pengenalan perangkat keras komputer, yang dirancang untuk mengatasi keterbatasan aksesibilitas, performa, sumber daya, dan lisensi dari platform pengembangan populer seperti Unity.

Adapun tujuan khusus penelitian ini untuk mewujudkan solusi tersebut meliputi:

- 1) Merancang arsitektur aplikasi berbasis Three.js dan WebXR yang mampu menghasilkan kinerja optimal pada mode *desktop* maupun perangkat VR.
- 2) Mengimplementasikan optimasi model 3D untuk menjaga performa aplikasi tetap stabil.
- 3) Membangun mekanisme interaksi yang konsisten dan mudah digunakan pada mode *desktop* dan VR dengan memanfaatkan fitur Three.js.

1.6 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

- 1) Bagi Universitas Ma Chung, menambah referensi akademik dan bahan ajar terkait teknologi Three.js dan WebXR, serta mendorong riset di bidang teknologi imersif.
- 2) Bagi penulis, memberikan pengalaman dalam merancang alur kerja pengembangan aplikasi VR, meningkatkan keahlian dalam penggunaan Three.js dan menghasilkan portofolio pribadi yang bermanfaat.
- 3) Bagi masyarakat, menyediakan media pengenalan interaktif untuk pengenalan perangkat keras komputer, menyediakan referensi alur kerja bagi pengembang atau mahasiswa lain yang tertarik pada teknologi serupa.

1.7 Luaran Penelitian

Penelitian ini diharapkan menghasilkan luaran sebagai berikut:

- 1) Sebuah aplikasi *Virtual Reality* (VR) pengenalan perangkat keras komputer berbasis web (Three.js dan WebXR) yang fungsionalitas, performa, dan usabilitasnya telah teruji dan tervalidasi, serta dapat diakses secara optimal baik pada mode *desktop* maupun *headset* VR.
- 2) Sebuah dokumentasi yang membahas tentang metodologi pengembangan, arsitektur teknis, optimasi performa, dan hasil pengujian dari aplikasi yang dibangun sehingga memenuhi standar publikasi dalam jurnal ilmiah.

1.8 Sistematika Penulisan

Untuk memberikan gambaran yang jelas mengenai struktur dan isi laporan tugas akhir ini, berikut adalah sistematika penulisan yang digunakan:

- 1) **Bab I Pendahuluan**, Bab ini berisi pendahuluan yang menjadi dasar penelitian, mencakup latar belakang masalah, identifikasi masalah, batasan masalah, perumusan masalah, tujuan yang ingin dicapai, manfaat yang diharapkan dari penelitian, serta luaran yang akan dihasilkan.
- 2) **Bab II Tinjauan Pustaka**, Bab ini menguraikan landasan teoretis dan studi terdahulu yang relevan. Bagian ini mencakup analisis penelitian sebelumnya untuk menemukan celah penelitian (*research gap*) serta pembahasan mendalam mengenai konsep dan teknologi yang digunakan, seperti *Virtual Reality* (VR), *WebGL*, *Three.js*, *WebXR*, dan metodologi *Multimedia Development Life Cycle* (MDLC).
- 3) **Bab III Analisis dan Perancangan Sistem**, Bab ini menjelaskan secara rinci metodologi penelitian yang digunakan. Dimulai dengan alur penelitian secara keseluruhan, dilanjutkan dengan analisis kebutuhan sistem yang terbagi menjadi kebutuhan fungsional, Non Fungsional, dan kebutuhan pengguna. Bagian utama bab ini adalah perancangan sistem menggunakan metode MDLC yang mencakup tahap konsep, perancangan (diagram *flowchart*, *use case*, *activity*, dan *storyboard*), pengumpulan materi, pembuatan, pengujian, hingga distribusi.
- 4) **Bab IV Hasil dan Pembahasan**, Bab ini menyajikan hasil implementasi dari seluruh rancangan. Mencakup hasil implementasi aplikasi, hasil optimasi aset 3D, serta hasil dari metode pengujian fungsionalitas (*black box*), performa (waktu muat dan FPS), dan usabilitas (SUS).
- 5) **Bab V Simpulan dan Saran**, Bab ini berisi simpulan akhir yang ditarik dari hasil analisis dan pembahasan. Simpulan ini menjawab rumusan masalah penelitian. Selain itu, bab ini juga menyertakan saran untuk pengembangan aplikasi di masa depan.

Bab II

Tinjauan Pustaka

2.1 Three.js

Menurut Straccia (2024) dalam bukunya “*Interactive Web Development With Three.js and A-Frame*”, Three.js diciptakan oleh Ricardo Cabello (Mr.doob) dan dirilis pertama kali pada tahun 2010. Three.js adalah sebuah pustaka JavaScript yang berfungsi sebagai penghubung antara kode JavaScript dengan API WebGL yang kompleks. WebGL, sebagai teknologi bawaan peramban modern, memungkinkan perenderan grafis 3D yang dipercepat oleh GPU tanpa memerlukan *plug-in* eksternal. Namun, pemrograman murni menggunakan WebGL sangat kompleks karena memerlukan penulisan kode *shader* dalam GLSL ES dan pengelolaan *vertex* secara manual. Three.js menyederhanakan proses ini secara signifikan, sehingga pengembang dapat membuat serta memanipulasi adegan 3D yang kompleks hanya dengan berbekal pengetahuan JavaScript.

1) Struktur dan Komponen Dasar Three.js

Struktur dasar sebuah aplikasi Three.js terdiri dari beberapa komponen fundamental yang secara bersama-sama menghasilkan visualisasi tiga dimensi (Straccia, 2024). Komponen pertama adalah adegan (*scene*), yang berfungsi sebagai wadah bagi seluruh objek, sumber cahaya, dan kamera yang akan digunakan. Selanjutnya, terdapat *renderer* yang bertugas mengambil data dari adegan dan menampilkannya pada elemen `<canvas>` dalam dokumen HTML.

Untuk dapat melihat objek dalam adegan, dibutuhkan sebuah kamera (*camera*), di mana jenis yang paling umum adalah *PerspectiveCamera* yang mensimulasikan mekanisme penglihatan manusia dengan efek distorsi perspektif. Objek 3D atau *mesh* terdiri dari dua komponen utama, yaitu geometri yang mendefinisikan bentuk objek, misalnya kubus, bola, atau model tiga dimensi yang diimpor, serta material yang mengatur properti visual permukaan objek, termasuk warna, tekstur, dan respons terhadap pencahayaan.

Selain itu, penerapan cahaya (*lights*) seperti *PointLight* atau *DirectionalLight* diperlukan untuk memberikan dimensi dan realisme pada objek, yang juga memungkinkan terbentuknya bayangan (*shadows*). Keseluruhan elemen ini dijalankan melalui *animation loop*, sebuah fungsi rekursif yang secara terus-menerus merender adegan pada setiap *frame*, sehingga memungkinkan terjadinya animasi dan interaksi secara *real-time*.

2) Interaktivitas, Material, dan Integrasi Aset pada Three.js

Three.js menyediakan berbagai mekanisme untuk mengintegrasikan interaktivitas ke dalam adegan tiga dimensi (Straccia, 2024). Interaksi dasar dapat diwujudkan melalui penerapan *event listener* pada perangkat *input* seperti *keyboard*, *mouse*, dan layar sentuh, yang memungkinkan manipulasi objek atau kontrol kamera secara langsung oleh pengguna. Untuk interaksi yang lebih kompleks, seperti deteksi objek yang dipilih (*clicked*), Three.js menggunakan metode *raycasting*, yaitu pemancaran sinar virtual dari posisi kamera menuju koordinat kursor pada layar untuk menentukan objek mana yang bersinggungan dengan sinar tersebut.

Untuk meningkatkan kualitas visual dan realisme hasil *render*, Three.js menggunakan material berbasis fisika (*Physically Based Rendering/PBR*), seperti *MeshStandardMaterial*. Material ini meniru sifat-sifat fisik dari benda di dunia nyata, seperti tingkat kemetalan (*metalness*) dan kekasaran permukaan (*roughness*), sehingga pencahayaan dan bayangan yang dihasilkan tampak lebih natural dan konsisten. Selain itu, Three.js juga memungkinkan untuk melakukan *import* aset tiga dimensi dari perangkat lunak pemodelan eksternal, seperti Blender, dengan berbagai format *file standar*, terutama format glTF dan glb yang direkomendasikan karena efisiensi dan kompatibilitasnya yang tinggi.

2.2 WebXR

WebXR (*Web Extended Reality*) menggabungkan teknologi *Virtual Reality* (VR) dan *Augmented Reality* (AR) ke dalam satu standar API yang dapat diakses langsung melalui peramban web tanpa perlu menginstal aplikasi tambahan. Dengan teknologi ini, pengguna dapat menikmati konten 3D interaktif di berbagai

perangkat, seperti *headset* VR, kacamata AR, *smartphone*, maupun komputer *desktop*. WebXR bekerja bersama pustaka WebGL untuk merender adegan 3D dan menyiapkan kamera virtual yang digunakan untuk menavigasi adegan tersebut. API WebXR mengatur *rendering* 3D serta *input* dari perangkat, seperti *headset* dan *controller*, sedangkan WebGL dan Three.js bertanggung jawab mengolah model serta grafis visual agar tampilan menjadi lebih realistik dan interaktif (Mouattalib et al., 2023).

2.3 *Virtual Reality* (VR)

Menurut De Luna (2022), *virtual reality* (VR) adalah simulasi berbasis komputer yang memungkinkan pengguna berinteraksi dengan lingkungan tiga dimensi yang dibuat secara digital. Teknologi ini dirancang untuk menciptakan pengalaman yang terasa nyata dengan memanfaatkan mekanisme persepsi manusia. Dalam bukunya “*Introduction to Virtual Reality*”, De Luna menguraikan secara sistematis mulai dari konsep dasar VR, sejarah perkembangannya, sistem pendukung yang digunakan, perbedaan dengan *augmented reality* (AR), serta dampaknya pada manusia. Selain itu, buku ini juga membahas berbagai aplikasi VR di berbagai bidang dan prospek pengembangannya di masa depan.

1) Fondasi, Sejarah, dan Klasifikasi VR

Perkembangan VR memiliki sejarah yang panjang, bahkan sebelum era teknologi digital saat ini. Konsep dasar VR dimulai dengan penemuan stereoskop oleh Sir Charles Wheatstone pada tahun 1838, yang menunjukkan bagaimana otak mampu menggabungkan dua gambar dua dimensi menjadi persepsi kedalaman tiga dimensi (De Luna, 2022). Pada tahun 1962, Morton Heilig mengembangkan Sensorama, sebuah perangkat yang mampu merangsang beberapa indera secara simultan untuk menciptakan pengalaman sinematik yang imersif. Istilah “*virtual reality*” sendiri mulai populer setelah diperkenalkan oleh Jaron Lanier pada tahun 1987. Kemunculan Oculus Rift pada tahun 2010, yang kemudian diakuisisi oleh Facebook, menjadi titik penting dalam mempercepat kemajuan teknologi VR modern. Berdasarkan tingkat imersi yang diberikan, VR dapat diklasifikasikan ke dalam beberapa jenis utama, di antaranya sebagai berikut:

- a. *Fully Immersive* VR, memberikan pengalaman yang paling lengkap dengan cara mengisolasi pengguna dari dunia nyata, biasanya menggunakan perangkat *Head-Mounted Display* (HMD) yang menutupi pandangan pengguna sepenuhnya.
- b. *Non-Immersive* VR, disajikan melalui layar komputer konvensional, seperti pada simulator *desktop*, di mana pengguna tetap sadar akan lingkungan fisiknya.
- c. *Collaborative* VR, yang memungkinkan banyak pengguna berinteraksi dalam satu dunia virtual secara bersamaan, dan *Web-based* VR, yang membangun dunia virtual di atas *platform* internet.

2) Arsitektur Sistem VR dan Teknologi Pendukung

Sistem VR terdiri dari dua komponen utama, yaitu perangkat keras (*hardware*) dan perangkat lunak (*software*). Perangkat keras meliputi VR *Engine*, yaitu komputer yang memproses grafis dan simulasi secara *real-time*, perangkat *input* seperti pelacak gerak kepala dan tangan (*trackers*), *controller*, atau sarung tangan data, serta perangkat output yang menstimulasi indera pengguna, misalnya *Head-Mounted Display* (HMD) untuk visual, *headphone* untuk audio, dan perangkat haptik untuk sensasi sentuhan (De Luna, 2022).

3) Persepsi Sensorik dalam VR

De Luna (2022) menjelaskan bahwa untuk menciptakan ilusi kenyataan yang meyakinkan, VR harus dapat menstimulasi berbagai persepsi sensorik manusia secara efektif. Penjelasan mengenai persepsi yang terlibat dalam VR dapat dibagi sebagai berikut:

- a. Persepsi visual adalah yang paling dominan dalam VR. Ilusi kedalaman tercipta melalui *stereopsis*, yaitu penyajian gambar yang sedikit berbeda untuk masing-masing mata. Cara ini membuat otak menggabungkan kedua gambar sehingga pengguna merasakan kedalaman dan ruang tiga dimensi.
- b. Persepsi suara dicapai dengan teknologi audio spasial atau suara 3D. Teknologi ini meniru cara telinga manusia melokalisasi sumber suara di lingkungan sekitar menggunakan *Head-Related Transfer*

Function (HRTF), sehingga suara terasa datang dari arah yang realistik dan imersif.

- c. Persepsi haptik memberikan sensasi sentuhan dan kontak fisik. Sensasi ini dihasilkan lewat getaran, tekanan, atau umpan balik fisik lainnya yang mampu meningkatkan rasa kehadiran (*presence*) dan membuat pengalaman VR terasa lebih nyata.

4) Perbedaan *Virtual Reality* (VR) dan *Augmented Reality* (AR)

Penting untuk memahami perbedaan antara *Virtual Reality* (VR) dan *Augmented Reality* (AR). Dalam VR, dunia nyata di sekitar pengguna diganti sepenuhnya dengan dunia digital yang dibuat komputer, sehingga pengguna merasa seperti berada di tempat lain dan terpisah dari lingkungan sebenarnya. Sedangkan dalam AR, objek atau informasi digital ditambahkan ke dunia nyata yang dilihat pengguna, sehingga pengguna tetap melihat dan berinteraksi dengan lingkungan sekitar mereka, hanya saja dengan tambahan elemen virtual. Singkatnya, AR menambahkan hal baru ke dunia nyata, sementara VR membawa pengguna masuk ke dunia baru yang sama sekali berbeda (De Luna, 2022).

5) Aplikasi, Manfaat, dan Masa Depan VR

Aplikasi VR kini tidak hanya untuk permainan dan hiburan saja. Di bidang pendidikan, VR memungkinkan siswa melakukan kunjungan lapangan virtual dan membantu mereka memahami konsep yang sulit dengan lebih mudah, sehingga membuat informasi lebih mudah diingat. Di sektor kesehatan, VR digunakan untuk pelatihan bedah tanpa risiko, terapi bagi pasien yang mengalami fobia atau PTSD, serta membantu mengurangi rasa nyeri. Selain itu, VR juga dipakai dalam pariwisata untuk tur virtual, di arsitektur untuk melihat desain secara nyata, dan di bisnis untuk membuat prototipe produk serta melatih karyawan dalam lingkungan yang aman namun realistik. Ke depannya, VR diprediksi akan semakin sering digunakan sebagai alat komunikasi dan kolaborasi, terutama untuk tim yang bekerja dari jarak jauh (De Luna, 2022).

2.4 Perangkat Keras Komputer (*Hardware*)

Menurut Erkamim (2023), perangkat keras (*hardware*) adalah bagian fisik dari sistem komputer yang terdiri atas komponen elektronik maupun mekanis. Komponen ini berperan penting dalam menjalankan instruksi, mengolah informasi, sekaligus menyimpan data. Sebagai salah satu elemen utama dalam bidang Teknologi Informasi (TI), perangkat keras menjadi fondasi utama bagi berfungsinya sistem komputer serta aplikasi yang digunakan di atasnya.

Adapun beberapa komponen utama perangkat keras beserta fungsinya antara lain sebagai berikut:

- 1) Monitor merupakan perangkat *output* yang digunakan untuk menampilkan hasil pemrosesan komputer dalam bentuk teks maupun gambar. Fungsinya adalah menyajikan informasi visual kepada pengguna.
- 2) *Keyboard* adalah perangkat *input* yang digunakan untuk memasukkan teks, angka, dan perintah ke dalam komputer. Fungsinya adalah menyampaikan instruksi pengguna ke sistem.
- 3) *Mouse* adalah perangkat input yang digunakan untuk menggerakkan kursor di layar serta memilih atau membuka objek dengan klik. Fungsinya adalah memudahkan navigasi dan interaksi dengan komputer.
- 4) *Motherboard* adalah papan sirkuit utama yang menjadi pusat penghubung seluruh komponen perangkat keras komputer. Pada papan ini terdapat slot dan konektor untuk CPU, RAM, kartu ekspansi, serta perangkat lainnya. Fungsinya adalah menyatukan dan mengoordinasikan kerja tiap komponen.
- 5) *Central Processing Unit* (CPU) adalah otak komputer yang mengeksekusi instruksi program, melakukan perhitungan, dan mengoordinasikan kerja perangkat keras lain. Dengan adanya inti prosesor (*core*) yang dapat bekerja paralel, CPU mampu meningkatkan kinerja sistem. Fungsinya adalah memproses instruksi, menjalankan program, dan mengendalikan operasi perangkat keras.

- 6) Memori adalah tempat penyimpanan data dan instruksi yang dibutuhkan CPU dalam menjalankan operasi. Terdapat dua jenis utama, yaitu RAM, yang menyimpan data sementara saat komputer aktif, dan ROM, yang berisi instruksi dasar untuk proses awal saat komputer dinyalakan. Fungsinya adalah menyediakan data dan instruksi bagi sistem agar dapat berjalan.
- 7) Kartu Grafis (GPU) adalah perangkat keras yang menghasilkan *output* visual ke monitor. Komponen ini memiliki prosesor grafis khusus yang mampu mengolah data gambar dan video dengan cepat. Fungsinya adalah menyajikan tampilan grafis berupa gambar maupun video.
- 8) Kartu jaringan adalah perangkat keras yang memungkinkan komputer terhubung ke jaringan, baik lokal (LAN) maupun internet. Kartu ini dapat berupa *Ethernet* untuk koneksi kabel atau Wi-Fi untuk koneksi nirkabel. Fungsinya adalah mengirim dan menerima data melalui jaringan.
- 9) *Storage* (HDD/SSD) merupakan media penyimpanan utama pada komputer. HDD bekerja dengan piringan magnetik untuk menyimpan data, sedangkan SSD menggunakan *chip* memori sehingga lebih cepat dan tahan lama. Fungsinya adalah menyimpan data, program, serta *file* secara permanen.
- 10) Printer adalah perangkat *output* yang digunakan untuk menghasilkan salinan fisik dari dokumen digital. Fungsinya adalah mencetak teks maupun gambar ke dalam bentuk kertas.
- 11) Flashdisk (*flash drive*) adalah perangkat penyimpanan *eksternal* berukuran kecil yang digunakan untuk menyimpan dan memindahkan data. Fungsinya adalah menyediakan media penyimpanan *portabel* yang mudah diakses di berbagai komputer.

2.5 HTML (*Hypertext Markup Language*)

```
<!DOCTYPE html>
<html>
<head>
    <title>Contoh Struktur Dasar HTML</title>
</head>
<body>
    <h1>Selamat Datang di Website Saya</h1>
    <p>Ini adalah paragraf pertama saya.</p>
</body>
</html>
```

Gambar 2. 1 Contoh Kode Struktur Dasar HTML

HTML (*Hypertext Markup Language*) adalah bahasa standar untuk membangun dan menyusun elemen pada halaman web melalui penggunaan *tag-tag* tertentu yang menentukan tampilan teks, gambar, tautan, maupun elemen multimedia. Menurut Kurniawan (2022), HTML berfungsi sebagai bahasa markah yang membentuk tata letak dasar dan makna konten dalam sebuah halaman. Setiap halaman web disusun berdasarkan kerangka standar HTML, sehingga memahami struktur dan fungsi tag menjadi penting dalam penataan elemen mulai dari judul hingga konten utama. Penjelasan mengenai struktur dasar dapat dilihat pada Gambar 2.1, sementara Tabel 2.1 memaparkan fungsi *tag-tag* fundamental yang menjadi fondasi penyusunan dokumen HTML di *browser*.

Tabel 2. 1 Deskripsi Struktur Dasar Tag HTML

No	Tag HTML	Fungsi dan Penjelasan
1	<!DOCTYPE html>	Tag deklarasi pertama yang menunjukkan dokumen ini menggunakan HTML dan membantu <i>browser</i> menampilkan halaman dengan benar.
2	<html>	Tag pembuka dan penutup dokumen HTML, menjadi elemen utama yang membungkus seluruh isi halaman.
3	<head>	Bagian yang berisi informasi teknis seperti judul halaman <title>, tautan ke <i>stylesheet</i> , <i>meta data</i> , dan <i>script</i> .
4	<title>	Tag yang menentukan judul halaman yang muncul di tab <i>browser</i> , penting untuk SEO dan kejelasan halaman.
5	<body>	Bagian yang berisi seluruh konten yang tampak di halaman web, seperti teks, gambar, tabel, tautan, dan elemen multimedia.

2.6 CSS (*Cascading Style Sheets*)

Cascading Style Sheet (CSS) adalah bahasa yang digunakan bersama dengan bahasa *markup*, seperti HTML dan XML. CSS digunakan untuk menentukan format dan membangun tampilan dalam suatu web supaya terlihat lebih menarik, seperti mengatur jenis *font* yang akan digunakan, mengubah latar belakang halaman, dan mengganti warna tulisan maupun bentuk suatu komponen pada halaman web (Kurniawan, 2022).

Untuk mengubah tampilan visual dari struktur HTML, CSS dapat diterapkan melalui beberapa metode yang berbeda. Pilihan metode ini akan memengaruhi efisiensi, kemudahan pemeliharaan, dan cakupan gaya yang diterapkan, mulai dari perubahan pada satu elemen spesifik hingga desain yang konsisten di seluruh situs web. Tabel 2.2 menguraikan tiga metode utama dalam penerapan CSS, masing-masing dengan kelebihan dan skenario penggunaan yang berbeda.

Tabel 2. 2 Jenis Penerapan CSS pada Dokumen HTML

No	Metode	Deskripsi
1	<i>External</i> CSS	Menggunakan file .css terpisah yang dihubungkan ke dokumen HTML dengan tag <code><link></code> pada bagian <code><head></code> . Satu file CSS dapat digunakan untuk banyak halaman, sehingga memudahkan pemeliharaan dan konsistensi desain.
2	<i>Internal</i> CSS	Ditulis langsung di dalam dokumen HTML dengan menempatkan kode CSS pada tag <code><style></code> di bagian <code><head></code> . Gaya hanya berlaku untuk halaman tersebut, sehingga sesuai untuk pengaturan tampilan khusus satu halaman.
3	<i>Inline</i> CSS	Aturan gaya ditulis langsung pada atribut <code>style</code> di elemen HTML tertentu. Efektif untuk perubahan cepat pada satu elemen, tetapi tidak efisien untuk situs web berskala besar yang membutuhkan konsistensi desain.

2.7 JavaScript

JavaScript adalah bahasa pemrograman yang pertama kali diperkenalkan pada tahun 1995 oleh Brendan Eich dengan tujuan menjadikan halaman web menjadi interaktif dan dinamis. JavaScript merupakan salah satu bahasa pemrograman sisi klien (*client-side programming language*), yang berarti kode dijalankan langsung di *browser* pengguna tanpa melalui server (Kurniawan, 2022).

Sama seperti CSS, JavaScript dapat diimplementasikan ke dalam dokumen HTML melalui beberapa metode.

Sama seperti CSS, implementasi JavaScript untuk menambahkan interaktivitas pada halaman web juga dapat dilakukan dengan beberapa cara. Metode yang dipilih berdampak pada keterbacaan kode, modularitas, dan kemampuan untuk digunakan kembali (*reusability*). Pengembang perlu memilih pendekatan yang paling sesuai dengan kompleksitas fungsionalitas yang dibangun. Tabel 2.3 menyajikan perbandingan tiga cara penulisan kode JavaScript dalam dokumen HTML, menjelaskan bagaimana setiap metode memengaruhi struktur dan pengelolaan proyek.

Tabel 2. 3 Jenis Penerapan JavaScript pada Dokumen HTML

No	Jenis Penulisan	Cara Penulisan	Keterangan
1	<i>Inline</i> JavaScript	Menulis kode JavaScript langsung pada atribut elemen HTML	Cocok untuk kode singkat, mudah tetapi kurang terstruktur, sering digunakan untuk interaksi sederhana langsung pada elemen HTML.
2	<i>Internal</i> JavaScript	Menulis kode JavaScript di dalam tag <code><script></code> pada dokumen HTML (biasanya di <code><head></code> atau sebelum <code></body></code>)	Ditempatkan dalam satu file dengan HTML, lebih terorganisir dibandingkan <i>inline</i> , baik untuk kode yang relatif singkat dan hanya dipakai pada satu halaman.
3	<i>External</i> JavaScript	Menulis kode JavaScript di file terpisah berekstensi .js dan menghubungkannya ke HTML menggunakan <code><script src="nama_file.js"></script></code>	Memisahkan kode JS dari HTML, memudahkan pengelolaan kode yang panjang, memungkinkan penggunaan ulang kode di banyak halaman.

2.8 WebGL

Menurut Zhuk dan Tetuev (2020), WebGL adalah sebuah *Application Programming Interface* (API) yang digunakan untuk merender grafis 3D secara *real-time*, dirancang khusus untuk web, dan terintegrasi langsung dalam *browser* modern. Teknologi ini memungkinkan visualisasi objek serta adegan 3D interaktif di dalam halaman HTML tanpa memerlukan *plug-in* tambahan. Program WebGL terdiri atas kode kontrol yang ditulis dalam JavaScript dan kode *shader* yang menggunakan GLSL ES (*OpenGL ES Shading Language*), sehingga memberikan kontrol penuh atas properti objek, seperti skala, posisi, rotasi, tekstur, dan warna.

Kinerja WebGL cenderung lebih rendah dibandingkan dengan aplikasi 3D *native* seperti Unity karena dijalankan di dalam *browser* dengan batasan teknis dan keamanan tertentu. Oleh karena itu, pengembang perlu membatasi kompleksitas model dan materi yang digunakan agar aplikasi tetap lancar dan responsif. Selain itu, aplikasi berbasis WebGL mengharuskan pengunduhan data melalui jaringan internet, sehingga pengguna akan mengonsumsi kuota internet lebih banyak dibandingkan aplikasi yang diinstal secara lokal.

Agar aplikasi berbasis WebGL dapat berjalan mulus di berbagai perangkat, terutama dengan spesifikasi terbatas, pengembang perlu memperhatikan beberapa aspek teknis yang memengaruhi performa aplikasi. Menurut Threkit (2025), beberapa faktor utama yang harus diperhatikan antara lain:

- 1) Ukuran Aset Total

Aset 3D sebaiknya dijaga di bawah 10 MB untuk memastikan waktu muat yang dapat diterima pengguna. Waktu muat yang sangat cepat dapat dicapai jika ukuran total berada di bawah 5 MB.

- 2) Kompleksitas Geometri (*Vertex Count*)

Jumlah *vertex*, yaitu titik sudut pembentuk model 3D, sebaiknya tidak melebihi 100.000 *vertex* per adegan. Setiap *vertex* harus diproses oleh GPU, sehingga jumlah yang berlebihan akan membebani kinerja perangkat.

- 3) Resolusi Tekstur

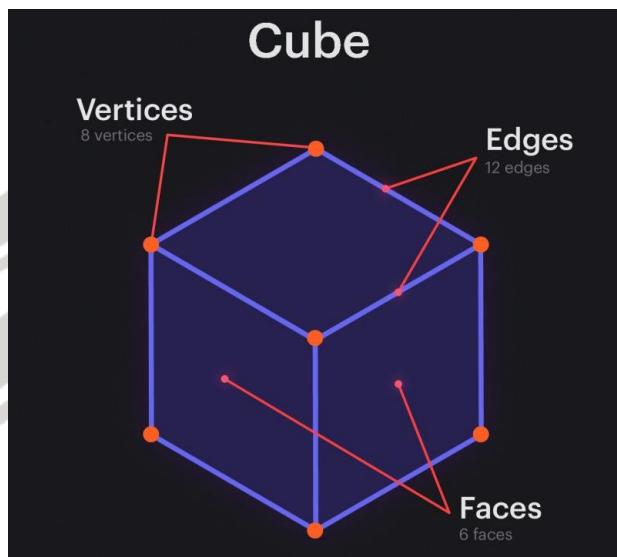
Penggunaan tekstur dengan resolusi 512×512 piksel lebih disarankan dibandingkan resolusi tinggi seperti 1024×1024 atau 2048×2048 . Tekstur akan didekompresi ke ukuran aslinya di memori video (VRAM), yang kapasitasnya terbatas.

- 4) *Draw Calls*

Jumlah objek independen yang digambar per *frame* (*draw calls*) idealnya dibatasi sekitar 200 per *frame* untuk mencapai kelancaran visual 30 fps. Menggabungkan beberapa objek dengan material yang sama merupakan salah satu strategi efektif untuk mengurangi beban *render*.

2.9 Pemodelan 3D (*3D Modeling*)

Pemodelan tiga dimensi atau *3D modeling* merupakan proses fundamental dalam grafis komputer untuk menciptakan representasi matematis dari suatu objek secara digital. Dalam penerapannya, pembentukan objek 3D tidak terlepas dari struktur geometri yang membangunnya. Mengacu pada artikel GameLab yang ditulis oleh Sani (2020), sebuah objek 3D atau yang sering disebut sebagai *mesh* sebenarnya merupakan satu kesatuan utuh yang tersusun dari kumpulan elemen-elemen geometri yang saling terhubung secara sistematis.



Gambar 2. 2 Struktur Dasar Objek 3D

Sumber: <https://cgtyphoon.com/fundamentals/vertices-edges-and-faces-of-3d-object/>
(diakses 5 Desember 2025)

Struktur pembentuk ini dimulai dari unit terkecil yang disebut *vertex*, yaitu sebuah titik tunggal yang posisinya didefinisikan oleh koordinat X, Y, dan Z dalam ruang virtual. Hubungan antar elemen ini kemudian membentuk struktur yang lebih kompleks. Sebagai contoh, ketika dua *vertex* disatukan maka akan terbentuk garis penghubung yang dikenal sebagai *edge*. Selanjutnya, apabila minimal tiga *vertex* dan *edge* terhubung membentuk suatu bidang permukaan berupa kurva tertutup, maka terbentuklah apa yang disebut dengan *face* (Sani, 2020). Pemahaman mendalam mengenai hierarki elemen pembentuk *mesh* ini, mulai dari titik, garis, hingga bidang, sangat krusial dalam penelitian ini terutama pada tahap optimasi aset untuk menjaga efisiensi kinerja aplikasi.

2.10 Blender

Menurut Moioli (2022), Blender merupakan perangkat lunak grafis komputer 3D yang bersifat *open source* dengan fungsi yang sangat luas. Perangkat lunak ini dapat dimanfaatkan pada berbagai bidang, termasuk animasi, efek visual, arsitektur, pencetakan 3D, serta desain produk. Keunggulan Blender terletak pada fleksibilitasnya yang mampu mengintegrasikan hampir seluruh proses produksi 3D dalam satu platform. Selain itu, Blender menyediakan dua mesin *render* utama, yaitu Cycles yang menghasilkan gambar dengan kualitas fotorealistik tinggi menggunakan teknik *path tracing*, sehingga mampu menampilkan pencahayaan realistik, bayangan halus, dan pantulan cahaya, serta Eevee yang memungkinkan proses *rendering* secara cepat dan *real-time*.

2.11 glTF-Transform

Berdasarkan dokumentasi resmi, glTF-Transform adalah sebuah perangkat lunak yang berfungsi untuk membaca, mengedit, dan menyimpan model 3D dalam format glTF 2.0. Dengan alat ini, pengguna dapat dengan mudah mengoptimalkan dan memodifikasi aset 3D melalui skrip, baik untuk penggunaan di komputer, *server*, maupun web. glTF-Transform mendukung berbagai fitur penting, seperti mengompresi *mesh* agar ukurannya lebih kecil, memperbaiki atau mengubah tekstur agar lebih efisien, dan mengedit detail seperti material atau tata letak objek. Selain itu, glTF-Transform juga bisa menggabungkan beberapa model menjadi satu, membuat tingkat detail yang berbeda untuk berbagai kebutuhan, dan menerapkan ekstensi khusus pada model. Dengan cara kerjanya yang memisahkan struktur data kompleks dari *file* biner, glTF-Transform memungkinkan pengubahan konten 3D dengan aman dan cepat, membuatnya sangat berguna bagi pengembang grafis 3D untuk memastikan aset mereka berkinerja dengan baik (McCurdy, 2025).

Pemanfaatan glTF-Transform dalam penelitian ini dilakukan melalui eksekusi perintah pada *Command Line Interface* (CLI). Setiap perintah memiliki fungsi spesifik yang menargetkan aspek optimasi yang berbeda, mulai dari kompresi geometri hingga penyederhanaan struktur aset. Tabel 2.4 berfungsi sebagai ringkasan praktis dari perintah-perintah kunci yang digunakan dalam alur kerja optimasi proyek ini, beserta fungsi utamanya.

Tabel 2. 4 Perintah CLI glTF-Transform dan Fungsinya

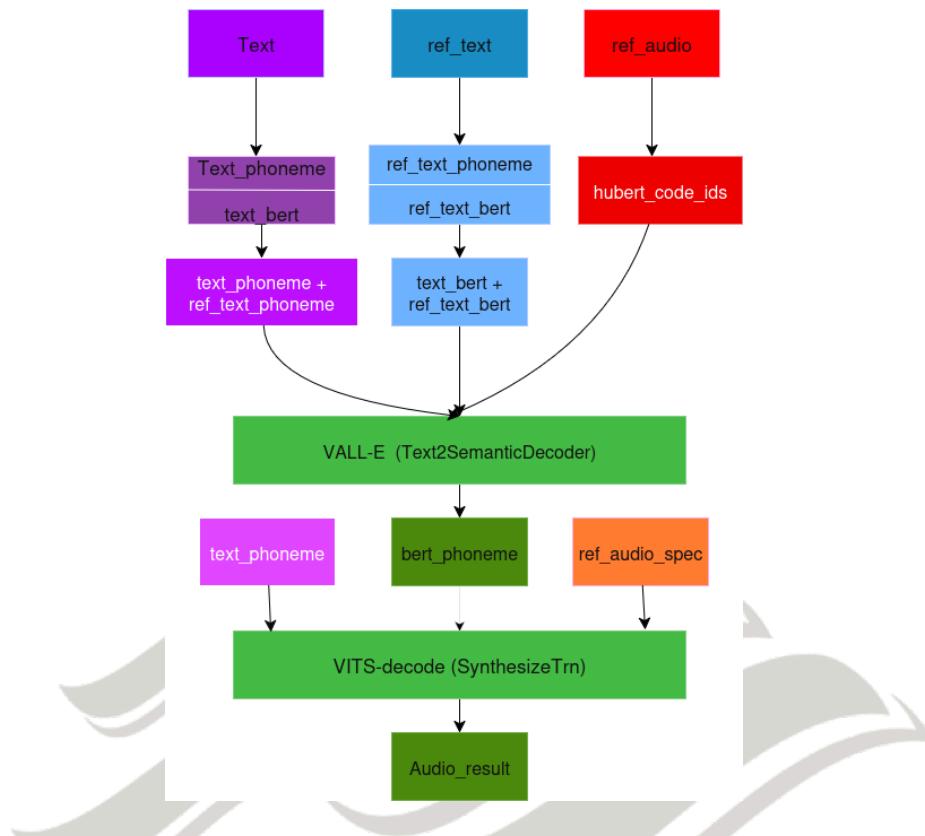
No	Perintah	Fungsi Utama	Contoh Penggunaan
1	<i>etc1s</i>	Mengompresi tekstur agar ukuran <i>file</i> lebih kecil.	gltf-transform etc1s input.glb output.glb
2	<i>optimize</i>	Membersihkan data duplikat, merapikan struktur, dan menyederhanakan <i>mesh</i> .	gltf-transform optimize input.glb output.glb
3	<i>draco</i>	Mengompresi geometri/ <i>mesh</i> untuk mengurangi ukuran <i>file</i> .	gltf-transform draco input.glb output.glb
4	<i>inspect</i>	Melihat struktur <i>file</i> .	gltf-transform inspect input.glb

2.12 glTF-Report

glTF-Report merupakan alat daring (*online tool*) yang berfungsi untuk menganalisis dan mengoptimalkan model 3D berformat glTF atau .glb. Terintegrasi dengan pustaka glTF Transform, alat ini memungkinkan pengguna mengevaluasi ukuran berkas (*disk size*), konsumsi memori grafis (VRAM), serta jumlah *draw call* yang dihasilkan oleh model (Meta, 2024). Melalui fitur tersebut, pengembang dapat mengidentifikasi asset yang berukuran besar, tekstur yang membebani GPU, maupun elemen geometri yang menyebabkan *overdraw*. Hasil analisis ini mendukung proses asset optimization agar model 3D tetap efisien, ringan, dan optimal digunakan pada aplikasi berbasis WebXR maupun VR.

2.13 GPT-SOVITS

GPT-SoVITS (*Generative Pre-trained Transformer - SoftVC VITS Singing Voice Conversion*) merupakan model sintesis suara berbasis *deep learning* yang mengadopsi pendekatan *few-shot learning*. Pendekatan ini memungkinkan model untuk meniru karakteristik vokal seorang penutur hanya dengan menggunakan data suara referensi berdurasi singkat, sekitar satu menit (Srivastava, 2025). Model ini adalah pengembangan dari proyek SOVITS yang diintegrasikan dengan arsitektur GPT guna meningkatkan kualitas, ekspresivitas, serta naturalitas suara yang dihasilkan. Proyek GPT-SoVITS bersifat *open source* dan dapat diakses melalui repositori GitHub resmi di bawah pengembang RVC-Boss, sehingga memungkinkan komunitas global untuk melakukan eksperimen dan pengembangan lanjutan (Zhang, et al., 2025; Srivastava, 2025).



Gambar 2. 3 Arsitektur Model GPT-SoVITS (Srivastava, 2025)

Sumber: <https://medium.com/@akshit0405/gpt-sovits-a-practical-guide-to-voice-synthesis-using-python-cce78818c933> (diakses 5 November 2025)

Secara umum, arsitektur GPT-SoVITS menggabungkan pendekatan hibrida antara *Transformer-based generative modeling* dan *system neural vocoder* berbasis VITS. Arsitektur ini tersusun atas empat komponen utama, yaitu Text Processing, Audio Processing, Komponen VALL-E, dan Modul Dekode VITS (Srivastava, 2025).

1) Modul Pemrosesan Teks (*Text Processing Module*)

Komponen ini berfungsi untuk mengubah teks mentah menjadi representasi fonemik yang bermakna. Proses tersebut dilakukan dengan memanfaatkan model BERT untuk melakukan penyandian (*encoding*) terhadap informasi fonemik. Hasil dari lapisan ketiga hingga terakhir pada BERT kemudian digunakan untuk menghasilkan tensor fonemik yang siap diproses pada tahap selanjutnya.

2) Modul Pemrosesan Audio (*Audio Processing Module*)

Modul ini bertanggung jawab untuk memproses audio referensi (*reference audio*) yang menjadi acuan karakteristik suara. Model CN-HuBERT digunakan untuk mengekstraksi `_ids` serta fitur frekuensi dari audio referensi. Tahapan ini memungkinkan sistem untuk menangkap karakteristik akustik seperti timbre, kualitas vokal, serta identitas penutur yang menjadi dasar utama dalam proses konversi suara.

3) Komponen VALL-E

Komponen yang dikembangkan oleh Microsoft Research ini berperan sebagai *sequence-to-sequence neural codec* model. VALL-E memperlakukan *text-to-speech* (TTS) sebagai tugas *conditional language modeling*, bukan sekadar regresi sinyal kontinu. Melalui pendekatan tersebut, model ini memiliki kemampuan *in-context learning*, sehingga dapat menghasilkan ucapan yang mempertahankan emosi dan konteks akustik dari audio referensi, bahkan hanya dengan sampel suara berdurasi tiga detik.

4) Modul Dekode VITS (VITS-Decode Module)

Modul ini mengimplementasikan arsitektur VITS (*Variational Inference with Adversarial Learning for End-to-End Text-to-Speech*) yang menggabungkan teknik *Variational Autoencoder* (VAE), *Generative Adversarial Network* (GAN), serta *normalizing flow*. Komponen ini mengintegrasikan *encoder* GlowTTS dan *vocoder* HiFi-GAN, menghasilkan suatu sistem *feed-forward* dengan performa waktu nyata yang dapat mencapai 67,12x faktor waktu-nyata pada GPU (Xu, 2024).

Keunggulan utama GPT-SoVITS terletak pada efisiensi data dan kualitas hasil suara. Pendekatan *few-shot learning* memungkinkan pelatihan model menggunakan data audio yang sangat terbatas, tanpa mengorbankan fidelitas dan ekspresivitas vokal. Integrasi modul SoftVC memungkinkan sistem untuk menangkap elemen prosodi, ritme, dan intonasi secara akurat. Sementara itu, dukungan VALL-E dan VITS menjamin hasil sintesis yang alami serta konsisten dengan konteks emosional dari penutur asli (Srivastava, 2025).

2.14 Github

GitHub didirikan oleh Tom Preston-Werner, Chris Wanstrath, PJ Hyett, dan Scott Chacon pada tahun 2008. GitHub merupakan suatu platform berbasis web yang dapat digunakan dalam pengembangan perangkat lunak sebagai pengelolaan kode program atau berkolaborasi dalam pengembangan proyek. GitHub menggunakan sistem kontrol versi Git, sehingga memungkinkan pengembang untuk melacak perubahan kode atau memudahkan pengembang ketika mengerjakan suatu proyek yang sama dalam kelompok (Deriota, 2023).

Dalam pengembangan perangkat lunak modern, platform seperti GitHub sangat penting untuk manajemen kode dan distribusi. Fitur-fitur yang disediakannya tidak hanya mendukung kerja kolaboratif, tetapi juga memfasilitasi proses penting seperti kontrol versi dan *hosting* aplikasi web. Tabel 2.5 menguraikan beberapa fitur utama GitHub yang relevan dan dimanfaatkan secara langsung dalam penelitian ini, terutama *repository* untuk pengelolaan kode dan GitHub Pages untuk distribusi aplikasi.

Tabel 2. 5 Fitur Utama GitHub

No	Fungsi	Penjelasan
1	<i>Repository</i> Git	Tempat penyimpanan proyek yang digunakan untuk mengelola semua perubahan pada direktori dan <i>file</i> dalam proyek.
2	Kolaborasi	Memungkinkan pengembang untuk meninjau, berdiskusi, dan menggabungkan kode dari berbagai kontributor dalam suatu tim.
3	Integrasi	GitHub dapat terhubung dengan berbagai perangkat lunak dan layanan lain seperti Visual Studio Code, Trello, Jira, dan Slack, sehingga mempermudah alur kerja pengembang.
4	<i>Version Control</i>	Sistem yang berfungsi melacak dan mencatat setiap perubahan kode program atau dokumen dalam proyek secara terstruktur dan terkontrol.
5	<i>GitHub Pages</i>	Layanan <i>hosting</i> halaman web statis secara gratis yang disediakan GitHub untuk mempublikasikan situs web langsung dari <i>repository</i> .

2.15 Cloudflare Pages

Cloudflare Pages merupakan layanan penerapan (*deployment*) berbasis *cloud* yang disediakan oleh Cloudflare untuk membangun serta mengelola aplikasi *full-stack* secara cepat melalui jaringan globalnya. Layanan ini memungkinkan pengembang untuk menerapkan proyek secara langsung dari repositori Git,

mengunggah berkas hasil kompilasi (*build*) secara manual melalui fitur *Direct Upload*, atau menggunakan antarmuka baris perintah (*Command Line Interface*) melalui Cloudflare C3 (Cloudflare Docs, 2025). Kemudian, salah satu keunggulan utama Cloudflare Pages adalah kemampuannya dalam menyediakan lingkungan *serverless* yang terintegrasi melalui Pages Functions. Fitur tersebut memungkinkan eksekusi kode pada sisi *server* tanpa memerlukan infrastruktur terpisah, sehingga pengembang dapat menambahkan logika dinamis pada aplikasi web secara efisien. Di samping itu, Cloudflare Pages juga menyediakan fitur *rollbacks* untuk memulihkan versi penerapan sebelumnya secara instan, serta fitur *redirects* untuk mempermudah pengelolaan rute dan pengalihan URL (Cloudflare Docs, 2025).

Cloudflare Pages memiliki integrasi erat dengan produk lain dalam ekosistem Cloudflare, seperti Workers yang berfungsi sebagai lingkungan eksekusi *serverless*, R2 Storage untuk penyimpanan objek tanpa biaya keluar (*egress fee*), D1 Database sebagai basis data *serverless* asli Cloudflare, serta Zaraz yang berperan dalam mengalihkan pemrosesan skrip pihak ketiga ke *cloud* guna meningkatkan kecepatan dan keamanan situs (Cloudflare Docs, 2025).

2.16 WebXR API Emulator

Menurut dokumentasi WebXR API Emulator (Mozilla Mixed Reality, 2024), WebXR API Emulator merupakan ekstensi *browser* yang dapat dimanfaatkan pengembang untuk mensimulasikan VR dan AR di *browser*. Emulator ini, memungkinkan pengembang untuk dapat menyesuaikan parameter posisi, rotasi, dan *input* perangkat, tanpa menggunakan perangkat khusus seperti *headset* VR atau AR, sehingga alat ini sangat bermanfaat dalam proses pengembangan dan pengujian.

2.17 Pengujian *Black Box*

Menurut Prasatya (2024) dalam artikel yang diterbitkan oleh CodePolitan, pengujian *black box* didefinisikan sebagai metode pengujian perangkat lunak yang memfokuskan evaluasi sepenuhnya pada aspek fungsionalitas aplikasi dari sudut pandang pengguna akhir. Dalam pendekatan ini, penguji memperlakukan perangkat lunak sebagai sebuah "kotak hitam", di mana pengujian dilakukan tanpa memedulikan pengetahuan mengenai struktur kode internal, logika pemrograman,

maupun arsitektur *back-end* yang membangunnya. Prinsip kerja utama dari metode ini berfokus pada pengujian kesesuaian antara input yang diberikan dengan output yang dihasilkan, di mana pengujian akan memasukkan berbagai kombinasi data ke dalam sistem dan memverifikasi apakah keluaran yang dihasilkan sudah sesuai dengan spesifikasi yang diharapkan tanpa perlu memahami algoritma yang berjalan di balik layar.

2.18 *System Usability Scale (SUS)*

Menurut artikel yang diterbitkan oleh School of Information Systems BINUS University (Andysa, 2022), *System Usability Scale (SUS)* merupakan sebuah kuesioner yang dapat digunakan untuk mengukur tingkat kegunaan (*usability*) dari suatu sistem komputer. Metode ini pertama kali diciptakan oleh John Brooke pada tahun 1986 dan telah menjadi standar industri karena karakteristiknya yang mudah digunakan, tidak memerlukan biaya besar, namun tetap mampu memberikan hasil yang valid dan reliabel. Rincian kesepuluh instrumen pernyataan yang digunakan dalam metode SUS dapat dilihat pada Tabel 2.6.

Tabel 2. 6 Instrumen Pernyataan *System Usability Scale (SUS)*

No	Pernyataan	Skor (1–5)
1	Saya berpikir akan sering menggunakan aplikasi ini.	
2	Saya merasa aplikasi ini rumit untuk digunakan.	
3	Saya merasa aplikasi ini mudah dipelajari.	
4	Saya memerlukan bantuan orang lain atau dukungan teknis untuk menggunakan aplikasi ini.	
5	Saya merasa berbagai fungsi dalam aplikasi ini terintegrasi dengan baik.	
6	Saya merasa ada terlalu banyak inkonsistensi dalam aplikasi ini	
7	Saya merasa banyak orang akan belajar menggunakan aplikasi ini dengan cepat.	
8	Saya merasa aplikasi ini membingungkan saat digunakan.	
9	Saya merasa percaya diri saat menggunakan aplikasi ini.	
10	Saya harus mempelajari banyak hal terlebih dahulu sebelum bisa menggunakan aplikasi ini.	

SUS dirancang untuk menilai persepsi pengguna terhadap efektivitas, efisiensi, dan kepuasan penggunaan sebuah aplikasi atau produk melalui 10 butir pernyataan yang dijawab menggunakan skala Likert 5 poin, mulai dari 1 sangat tidak setuju, 2 tidak setuju, 3 netral, 4 setuju, dan 5 sangat setuju. Struktur

pertanyaannya disusun secara selang-seling antara pernyataan bernuansa positif dan negatif untuk menghindari bias jawaban, sementara perhitungan skor akhir SUS mengikuti mekanisme penilaian khusus yang telah ditetapkan sebagai berikut:

- 1) Pernyataan pada nomor ganjil (1, 3, 5, 7, 9): Merupakan pernyataan positif. Skor kontribusi dihitung dengan mengurangi nilai jawaban pengguna dengan angka 1 ($\text{Skor} - 1$).
- 2) Pernyataan pada nomor genap (2, 4, 6, 8, 10): Merupakan pernyataan negatif. Skor kontribusi dihitung dengan mengurangi angka 5 dengan nilai jawaban pengguna ($5 - \text{Skor}$).

Setelah mendapatkan skor kontribusi dari setiap pertanyaan, seluruh skor dijumlahkan dan dikalikan dengan bilangan konstanta 2,5. Proses ini akan menghasilkan skor akhir SUS dalam rentang 0 hingga 100.

Penting untuk dipahami bahwa skor SUS bukanlah persentase, melainkan skor persentil. Berdasarkan pedoman interpretasi umum, skor di atas 68 dianggap berada di atas rata-rata (*average*), sementara skor di bawah 68 mengindikasikan perlunya perbaikan pada aspek usabilitas sistem. Penggunaan metode ini sangat relevan untuk penelitian yang melibatkan pengujian pengalaman pengguna (*user experience*) karena kemampuannya dalam menarik kesimpulan yang kuat meskipun dengan jumlah sampel responden yang relatif kecil.

2.19 Penelitian Terdahulu

Untuk memosisikan penelitian ini dalam lanskap ilmiah yang ada, dilakukan analisis mendalam terhadap studi-studi relevan sebelumnya. Analisis ini berfokus pada identifikasi kontribusi utama setiap penelitian serta celah (*research gap*) yang ada, sehingga oriinalitas dan kontribusi penelitian ini dapat ditunjukkan dengan jelas. Tabel 2.7 menyajikan ringkasan dari analisis perbandingan tersebut, memetakan setiap *research gap* dengan solusi spesifik yang ditawarkan dalam penelitian ini.

Tabel 2. 7 Analisis Keterbatasan Studi Terdahulu

No.	Peneliti (Tahun)	Kontribusi Utama	Keterbatasan	Solusi Penelitian Ini
1	Astianingrum dkk. (2025)	Membuktikan efektivitas Three.js dan WebGL untuk visualisasi 3D yang dapat dijalankan pada berbagai perangkat.	Belum membahas integrasi mode VR.	Menambahkan mode VR dengan dukungan WebXR.
2	Matahari (2022)	Menerapkan pengembangan aplikasi VR berbasis WebXR.	Belum mencakup langkah optimasi aset, sehingga performa tidak dibahas menyeluruh.	Menambahkan tahap optimasi.
3	Putra dkk., (2022)	Membuktikan <i>Vertex Decimation</i> dapat mengurangi kompleksitas model hingga 26,7%.	Optimasi hanya pada tahap pemodelan dan belum diuji pada aplikasi nyata.	Menambahkan optimasi lebih lanjut dengan menggunakan alat glTF-Transform.
4	Kurniasari dkk., (2023); Fitriyanto dkk., (2024); Anggoro dkk., (2024)	Memvalidasi efektivitas metode MDLC dalam menghasilkan aplikasi AR/VR dengan penerimaan >70%.	Fokus pada aplikasi <i>native</i> (Unity 3D), belum membahas pengembangan berbasis web.	Mengadopsi metode MDLC untuk pengembangan berbasis web.

1) Implementasi Three.js

Astianingrum dkk. (2025) dalam penelitiannya yang berjudul “*Proses Rendering 3D Model Jantung Manusia pada Aplikasi Mobile Web Menggunakan WebGL*”. Hasil dari penelitian ini menunjukkan bahwa *rendering* model 3D berbasis WebGL dengan format GLB (.glb) dapat berjalan dengan baik pada perangkat *desktop* maupun *mobile* tanpa memerlukan instalasi tambahan. Hal ini membuktikan efektivitas Three.js dan WebGL untuk visualisasi 3D yang dapat diakses melalui berbagai perangkat.

2) Implementasi WebXR

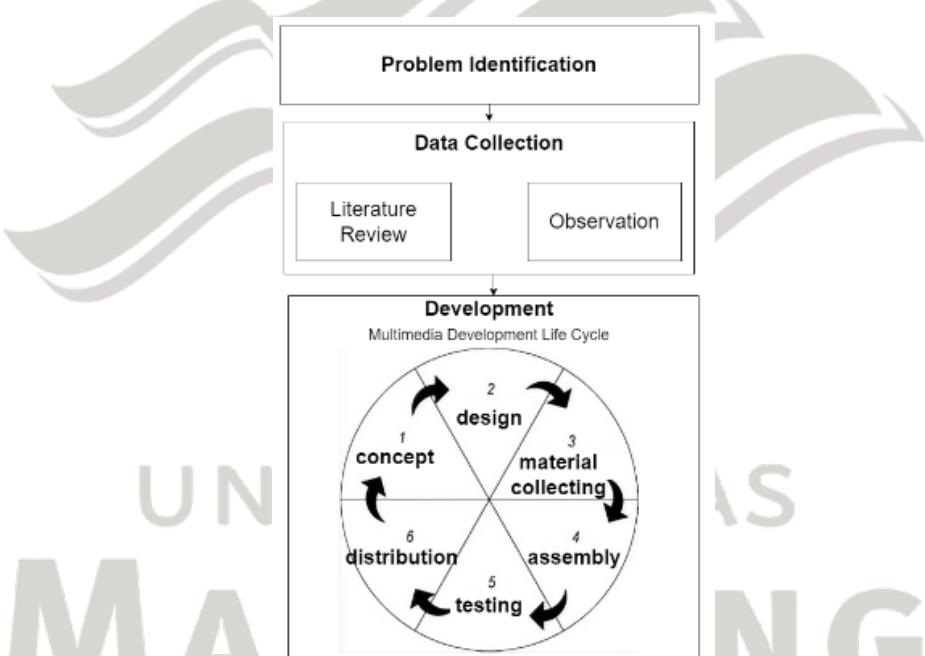
Matahari (2022) dalam penelitiannya yang berjudul “*WebXR Asset Management in Developing Virtual Reality Learning Media*”. Hasil penelitian ini menunjukkan bahwa WebXR sebagai API berbasis JavaScript

memungkinkan pengalaman imersif dapat diakses langsung melalui *browser* web, sehingga mengurangi hambatan aksesibilitas. Penelitian ini memberikan kontribusi penting terkait alur kerja dan manajemen aset dalam pengembangan aplikasi WebXR.

3) Optimasi Aset 3D

Putra dkk. (2022) dalam penelitiannya yang berjudul “*Optimasi Aset dan Karakter Permainan 3D Berbasis Tematik Sekolah Dasar*”. Hasil penelitian ini membuktikan bahwa metode *Vertex Decimation* dapat secara efektif mengurangi kompleksitas model 3D hingga 26,7% dari jumlah total *vertex* tanpa menyebabkan kehilangan detail yang signifikan.

4) Metode *Multimedia Development Life Cycle* (MDLC)



Gambar 2.4 Contoh Alur Penelitian dengan MDLC (Anggoro dkk., 2024)

Gambar 2.4 menyajikan contoh alur penelitian untuk pengembangan aplikasi realitas virtual yang menggunakan metode MDLC, diadaptasi dari penelitian oleh Anggoro dkk., (2024). Studi tersebut menerapkan enam tahapan MDLC yang meliputi Konsep, Desain, Pengumpulan Materi, Perakitan, Pengujian, dan Distribusi. Seluruh tahapan tersebut diaplikasikan untuk mengembangkan sebuah aplikasi *Virtual Reality Exposure Therapy* (VRET) yang ditujukan bagi penderita arachnophobia.

Selain itu, Fitriyanto dkk., (2024) dan Kurniasari dkk., (2023) dalam penelitian terkait implementasi metode MDLC. Hasil penelitian keduanya memvalidasi efektivitas metode *Multimedia Development Life Cycle* (MDLC) dalam menghasilkan aplikasi AR/VR, dengan tingkat penerimaan pengguna yang mencapai lebih dari 70%. Namun, fokus implementasi pada kedua penelitian tersebut adalah pada aplikasi *native* yang dikembangkan menggunakan Unity 3D.

Perbedaan utama penelitian ini dengan kajian sebelumnya adalah kelengkapan solusi yang ditawarkan. Penelitian ini mengintegrasikan visualisasi 3D berbasis web, implementasi VR dengan WebXR, dan metode optimasi asset yang komprehensif ke dalam satu alur kerja utuh. Berbeda dari penelitian lain yang berfokus pada satu aspek, penelitian ini menyajikan panduan lengkap dari optimasi hingga implementasi untuk menghasilkan aplikasi VR berbasis web yang optimal dan interaktif, serta mengadaptasi metode MDLC agar sesuai dengan konteks pengembangan yang dilakukan. Persamaan mendasar terletak pada penggunaan teknologi inti seperti Three.js dan WebXR, yang dikembangkan lebih lanjut dalam penelitian ini.

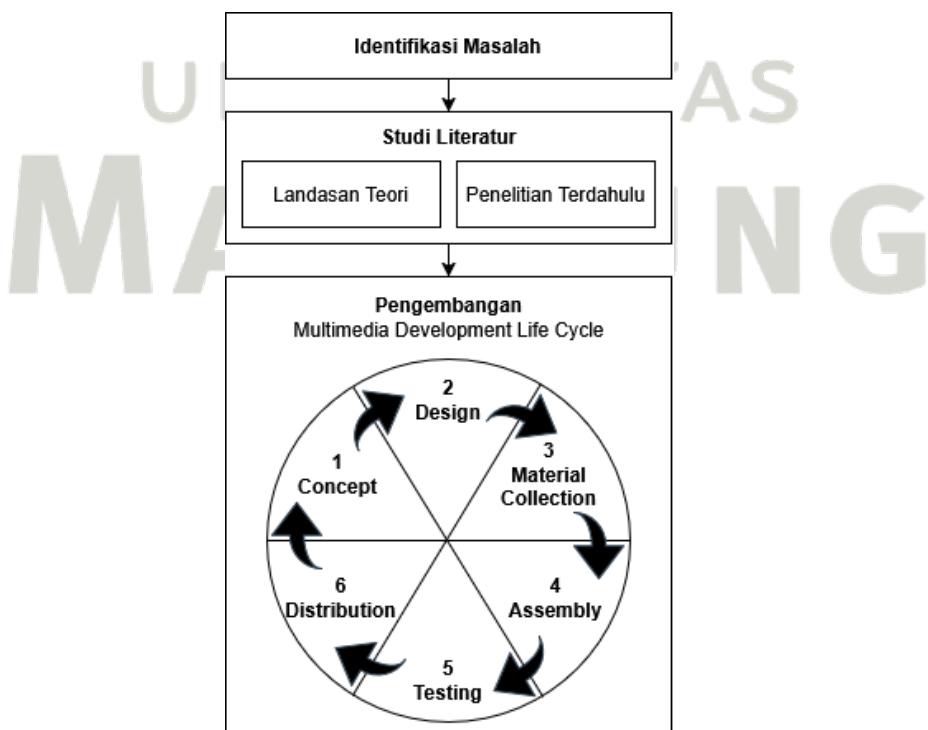
UNIVERSITAS
MA CHUNG

Bab III

Analisis dan Perancangan Sistem

3.1 Alur Penelitian

Bab ini menjelaskan tahapan yang digunakan dalam penelitian. Secara umum, alur penelitian terdiri dari tiga tahap utama, yaitu identifikasi masalah, studi literatur, dan pengembangan sistem. Tahap pertama mengacu pada pembahasan di Bab 1, yang berfokus pada analisis permasalahan yang ada, khususnya kesenjangan antara media pembelajaran konvensional yang bersifat statis dan kendala teknis pada platform VR seperti Unity. Tahap kedua mengacu pada Bab 2, di mana dilakukan untuk mengumpulkan referensi dan teori pendukung serta mengidentifikasi celah penelitian yang relevan. Tahap terakhir merupakan proses pengembangan sistem, yang menjadi inti dari penelitian ini. Pada tahap ini digunakan metode *Multimedia Development Life Cycle* (MDLC), yang meliputi tahapan konsep, perancangan, pengumpulan materi, pembuatan, pengujian, dan distribusi. Alur penelitian secara keseluruhan dapat dilihat pada Gambar 3.1.



Gambar 3. 1 Alur Penelitian

3.2 Concept (Konsep)

Tahap konsep merupakan dasar dari proses pengembangan aplikasi yang berfungsi untuk menetapkan arah, tujuan, dan ruang lingkup proyek. Pada tahap ini, ide utama dirumuskan, target pengguna ditentukan, serta kebutuhan sistem, baik Fungsional maupun Non Fungsional termasuk kebutuhan perangkat keras dan perangkat lunak yang dijabarkan secara rinci sebagai acuan untuk tahap perancangan dan pengembangan berikutnya.

3.2.1 Ide dan Tujuan Aplikasi

Ide utama penelitian ini adalah mengembangkan media pembelajaran berbasis *Virtual Reality* (VR) sebagai alternatif dari platform pengembangan Unity dengan memanfaatkan teknologi web modern, yaitu Three.js dan WebXR. Tujuannya adalah merancang aplikasi pembelajaran perangkat keras komputer yang interaktif, ringan, dan dapat diakses langsung melalui *browser*. Fokus pengembangannya terletak pada optimalisasi aset 3D serta perancangan mekanisme interaksi yang konsisten antara perangkat VR dan *desktop* untuk mengevaluasi performa, kenyamanan, dan penerimaan pengguna.

3.2.2 Target Pengguna

Aplikasi ini ditujukan bagi pelajar atau mahasiswa di bidang teknologi informasi serta masyarakat umum yang memiliki minat awal terhadap perangkat keras komputer. Melalui aplikasi ini, pengguna diharapkan dapat mempelajari dan memahami berbagai komponen komputer secara interaktif, sehingga dapat menambah wawasan dasar tentang struktur dan fungsi perangkat keras.

3.2.3 Analisis Kebutuhan Fungsional

Untuk memenuhi kebutuhan pengguna dan mencapai tujuan pengembangan aplikasi, serangkaian kebutuhan Fungsional telah didefinisikan. Kebutuhan ini menjelaskan fitur-fitur spesifik serta kemampuan yang harus dimiliki oleh sistem, antara lain sebagai berikut:

- 1) Sistem dapat menampilkan model 3D komponen perangkat keras utama komputer, seperti monitor, *keyboard*, *mouse*, *motherboard*, CPU, RAM, GPU, dan lainnya.

- 2) Sistem menyediakan dua mode operasi, yaitu mode *desktop* menggunakan *mouse* dan mode VR untuk *headset* yang mendukung WebXR.
- 3) Sistem menyediakan mekanisme interaksi dasar, seperti kontrol kamera dan memutar model 3D.
- 4) Sistem menyajikan panel informasi yang berisi deskripsi untuk setiap komponen perangkat keras.
- 5) Sistem memiliki fitur penjelasan audio untuk setiap materi komponen.
- 6) Sistem menyediakan antarmuka navigasi untuk berpindah antar materi pembelajaran komponen secara berurutan.
- 7) Sistem menyertakan mini kuis benar/salah setelah pengguna menyelesaikan satu materi untuk membuka materi berikutnya.
- 8) Sistem menyediakan tes akhir *multiple choice* setelah semua materi berhasil diselesaikan oleh pengguna.
- 9) Sistem dapat menampilkan laporan belajar yang berisi skor akhir dari tes yang telah dikerjakan.

3.2.4 Analisis Kebutuhan Non Fungsional

Subbab ini menguraikan kebutuhan Non Fungsional yang mencakup spesifikasi perangkat keras dan perangkat lunak pendukung pengembangan serta pengujian sistem.

- 1) Kebutuhan perangkat keras.

Proses pembuatan dan pengujian aplikasi VR interaktif ini memerlukan dukungan perangkat keras yang memadai. Spesifikasi perangkat keras yang digunakan dalam penelitian ini adalah sebagai berikut:

- a. Laptop dengan *processor* i7-4700, RAM 16GB, SSD 512GB
- b. *Headset* VR Oculus Quest 2

- 2) Kebutuhan perangkat lunak

Pada sisi perangkat lunak, penelitian ini memanfaatkan berbagai alat dan platform untuk membangun, mengoptimalkan, dan menyebarkan aplikasi. Rincian perangkat lunak yang dibutuhkan adalah sebagai berikut:

- a. Three.js dan WebXR berfungsi sebagai pustaka utama yang digunakan untuk melakukan *rendering* grafis tiga dimensi serta mendukung fungsionalitas *Virtual Reality* (VR) berbasis web.
- b. *Browser* berperan sebagai lingkungan eksekusi atau runtime tempat aplikasi web dijalankan.
- c. Sketchfab atau platform sejenis bersama Blender digunakan untuk membuat, mengedit, dan mengambil aset tiga dimensi yang akan digunakan dalam pengembangan aplikasi.
- d. GPT-SOVITS berfungsi untuk melakukan sintesis suara sehingga dapat menghasilkan narasi audio.
- e. glTF-Transform dan glTF Report digunakan untuk melakukan kompresi, modifikasi, serta analisis performa terhadap model tiga dimensi agar lebih efisien ketika dijalankan di lingkungan web.
- f. Visual Studio Code dan GitHub digunakan untuk menulis kode dan mengelola versi proyek agar proses pengembangan lebih teratur dan terkoordinasi.
- g. WebXR API Emulator berfungsi untuk melakukan pengujian terhadap fungsionalitas VR tanpa memerlukan perangkat VR fisik.
- h. Cloudflare Pages digunakan sebagai platform untuk melakukan hosting dan distribusi aplikasi berbasis web secara cepat dan aman.

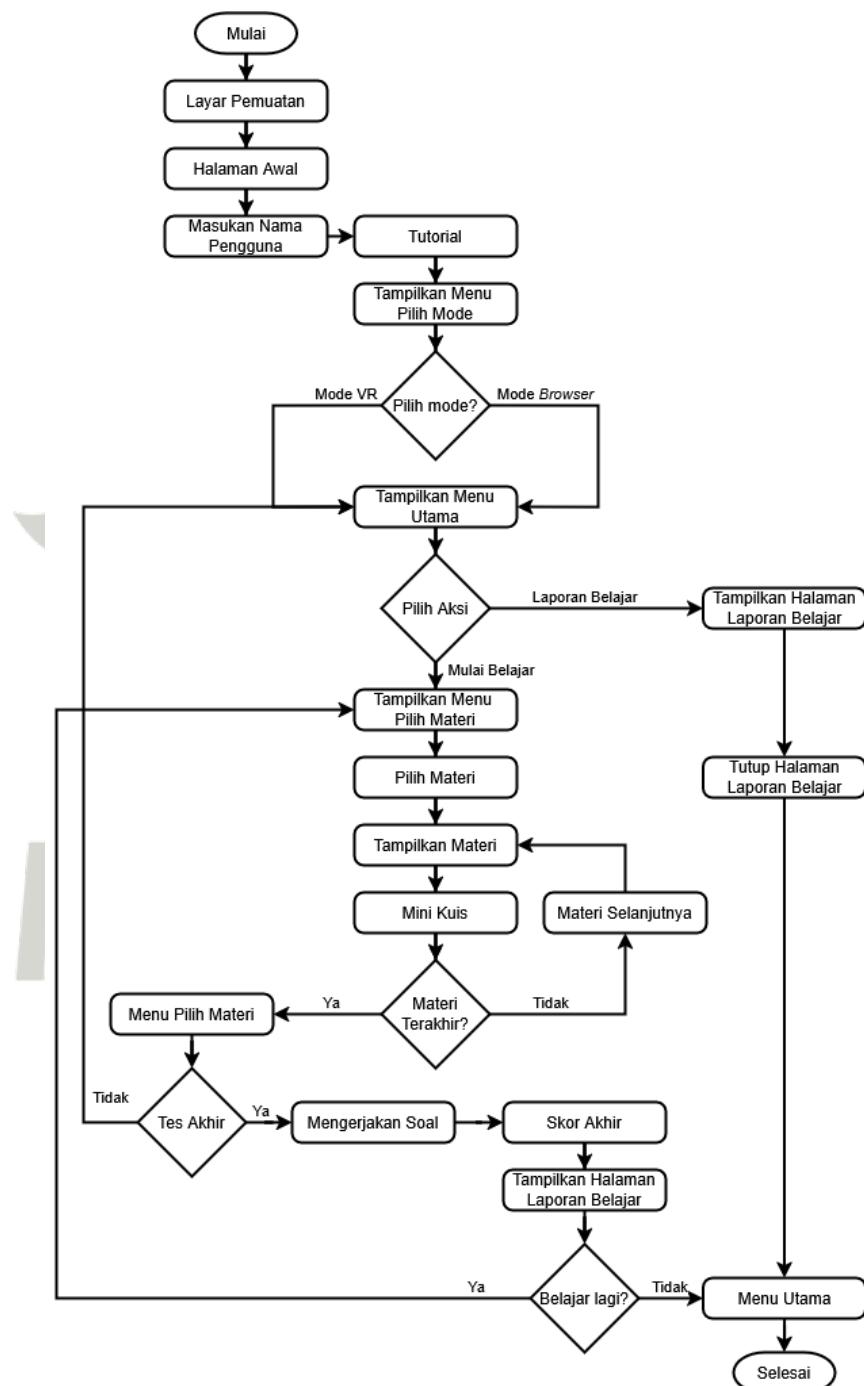
3.3 **Design** (Perancangan)

Tahap perancangan merupakan fase di mana konsep yang telah didefinisikan diwujudkan ke dalam spesifikasi teknis dan visual yang lebih rinci. Pada tahap ini, alur kerja sistem, interaksi fungsional, dan pengalaman pengguna dirancang secara mendetail. Untuk memvisualisasikan rancangan tersebut, digunakan beberapa pemodelan, yaitu diagram *flowchart* untuk menggambarkan alur proses secara keseluruhan, diagram *use case* untuk mendefinisikan interaksi antara pengguna dan sistem, diagram *activity* untuk memodelkan alur kerja spesifik, dan *Storyboard* untuk merancang antarmuka serta pengalaman visual pengguna.

3.3.1 **Diagram Flowchart**

Untuk menggambarkan alur proses dan interaksi pengguna dengan sistem secara logis, digunakan diagram *flowchart*. Seperti yang diperlihatkan pada Gambar

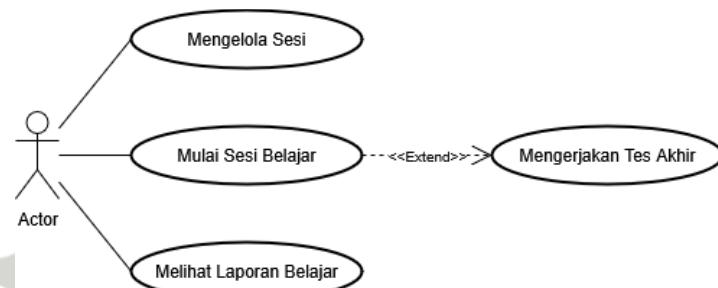
3.2, diagram alur aplikasi ini merinci proses kerja mulai dari tahap inisialisasi (memasukkan nama pengguna dan tutorial), dilanjutkan dengan pilih mode, sesi belajar interaktif yang berisi materi dan kuis, hingga diakhiri dengan evaluasi melalui tes akhir dan penampilan laporan hasil belajar.



Gambar 3. 2 Diagram Flowchart

3.3.2 Diagram Use Case

Untuk mendefinisikan fungsionalitas sistem dari perspektif pengguna, dibuatlah diagram *use case*. Diagram ini mengidentifikasi *actor* (pengguna) dan interaksi utama yang dapat mereka lakukan dengan aplikasi. Selanjutnya, setiap *use case* dirinci lebih lanjut menggunakan diagram *activity* untuk memodelkan langkah-langkah kerja dari awal hingga akhir.



Gambar 3. 3 Diagram *Use Case* Aplikasi

Setelah menyusun diagram *use case* untuk aplikasi berbasis WebXR, Tabel 3.1 menyajikan rincian setiap *use case*. Tabel ini memuat deskripsi dan tujuan pengguna, sehingga memfasilitasi pemahaman pembaca terhadap alur interaksi sistem yang diperlihatkan dalam diagram tersebut.

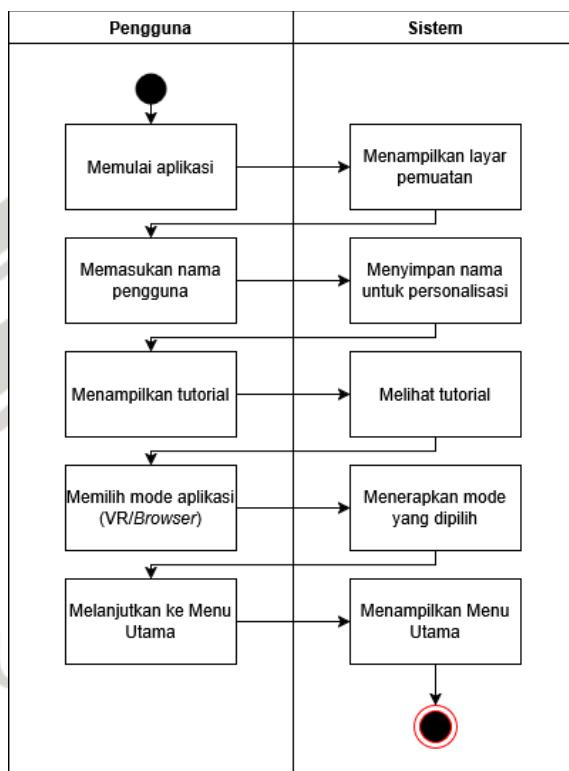
Tabel 3. 1 Deskripsi *Use Case* Aplikasi

No	Use Case	Deskripsi	Tujuan
1	Mengelola Sesi	Proses awal sebelum masuk ke menu utama.	Memberikan tutorial singkat mengenai cara penggunaan aplikasi serta menyiapkan sesi belajar sesuai preferensi pengguna (<i>desktop</i> atau perangkat VR).
2	Mulai Sesi Belajar	Pengguna memulai sesi belajar.	Pengguna memulai sesi belajar yang terdiri dari mempelajari materi dan mengerjakan mini kuis secara berurutan untuk setiap materi.
3	Mengerjakan Tes Akhir	Pengguna menyelesaikan tes akhir	Semua materi telah selesai dikerjakan, pengguna menyelesaikan tes akhir untuk mengukur pemahaman akhir.
4	Melihat Laporan Belajar	Pengguna melihat progres belajar.	Pengguna melihat progres dan kemajuan belajar.

3.3.3 Diagram Activity

Diagram aktivitas disajikan dalam format *swimlane*, dengan kolom terpisah untuk Pengguna dan Sistem, sehingga memudahkan pembaca untuk memahami peran masing-masing aktor dan alur proses yang terjadi. Setiap tabel fokus pada satu *use case*, sehingga dokumentasi tetap ringkas namun tetap mendetail untuk kebutuhan analisis dan desain sistem.

1) Diagram Activity Mengelola Sesi

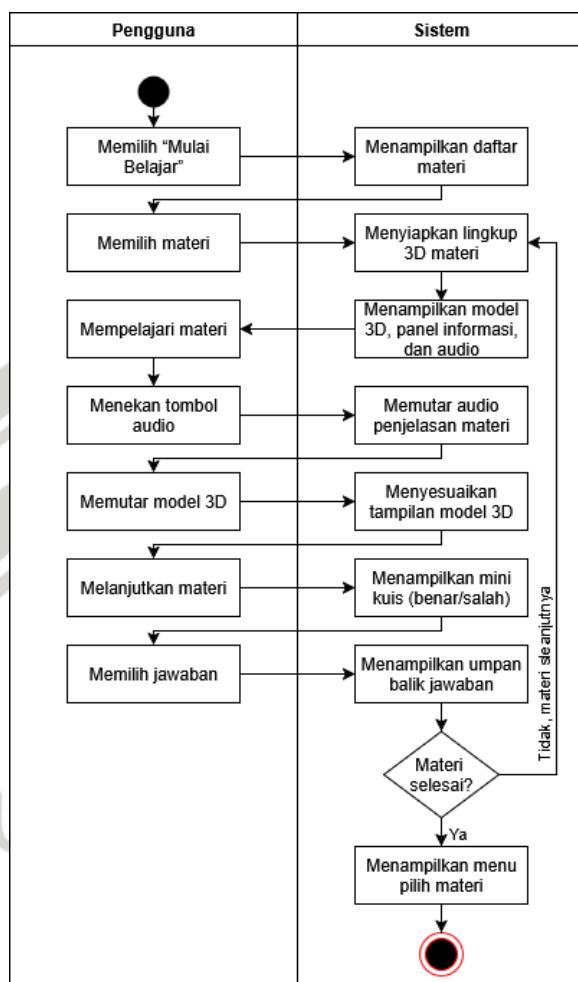


Gambar 3. 4 Diagram Activity Mengelola Sesi

Diagram *activity* swimlane tersebut menggambarkan interaksi antara pengguna dan sistem dalam alur penggunaan aplikasi. Pada tahap awal, pengguna memulai aplikasi yang kemudian menampilkan layar pemuatan sebagai tanda proses inisialisasi sedang berlangsung. Selanjutnya, pengguna diminta untuk memasukkan nama, yang kemudian dicatat dan disimpan oleh sistem untuk tujuan personalisasi. Setelah itu, sistem menampilkan tutorial singkat mengenai cara navigasi dan penggunaan aplikasi, sehingga pengguna lebih mudah memahami

alur interaksi. Sebelum masuk ke sesi belajar, pengguna memilih mode aplikasi, apakah menggunakan VR atau *browser*, dan sistem akan menyesuaikan antarmuka sesuai dengan mode yang dipilih.

2) Mulai Sesi Belajar

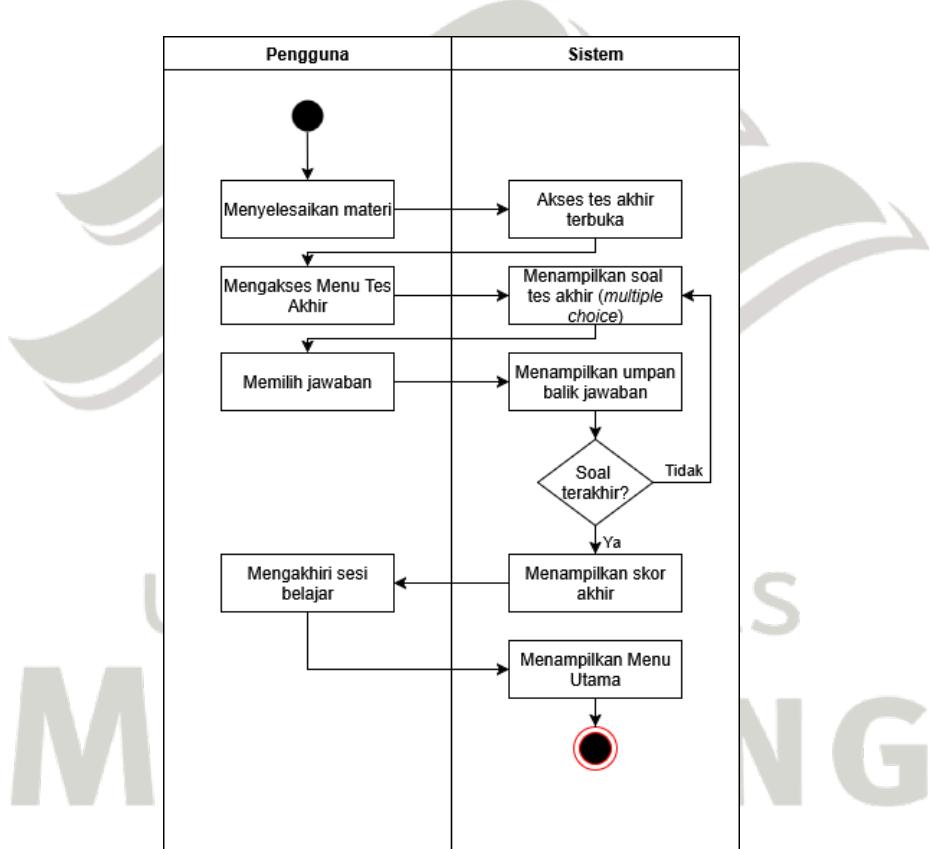


Gambar 3. 5 Diagram *Activity* Mulai Sesi Belajar

Diagram aktivitas ini menjelaskan proses sesi belajar yang dimulai oleh pengguna dengan memilih opsi "Mulai Belajar". Sistem kemudian menampilkan daftar materi yang tersedia untuk dipilih. Setelah pengguna memilih materi, sistem menyiapkan dan mengarahkan pengguna ke lingkungan 3D yang sesuai dengan materi tersebut. Pengguna dapat melihat model 3D, panel informasi, serta mendengarkan penjelasan audio terkait materi. Selama proses

pembelajaran, sistem memantau progres pengguna dan menyediakan fitur interaksi 3D, seperti kontrol kamera dan memutar model 3D. Setiap interaksi yang dilakukan pengguna, sistem akan memperbarui tampilan secara *real-time*. Selain itu, pengguna juga dapat mengerjakan mini kuis yang disediakan, dimana sistem akan menilai jawaban dan menampilkan skor sebagai umpan balik dan pengguna dapat melanjutkan ke sesi tes akhir setelah semua materi selesai dipelajari.

3) Mengerjakan Tes Akhir

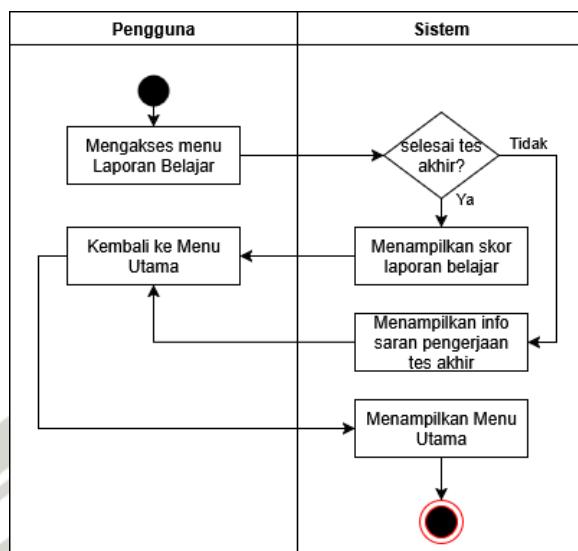


Gambar 3. 6 Diagram *Activity* Mengerjakan Tes Akhir

Diagram aktivitas ini menggambarkan proses pengambilan tes akhir yang dilakukan oleh pengguna setelah menyelesaikan seluruh materi pembelajaran. Setelah seluruh materi diselesaikan, sistem memberikan akses untuk mengikuti tes akhir. Pengguna mengerjakan soal yang disajikan, dan setelah selesai, sistem melakukan penilaian jawaban serta menyimpan skor hasil akhir sebagai rekaman capaian

pembelajaran pengguna. Setelah itu, pengguna diarahkan kembali ke menu utama.

4) Melihat Laporan Belajar



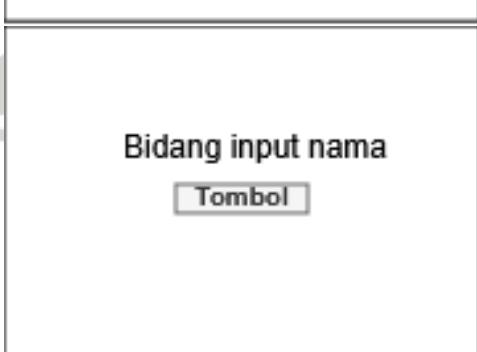
Gambar 3. 7 Diagram *Activity* Melihat Laporan Belajar

Diagram aktivitas ini menjelaskan proses ketika pengguna mengakses menu Laporan Belajar pada aplikasi. Sistem kemudian menampilkan ringkasan progres pembelajaran, yaitu skor hasil tes akhir yang telah dikerjakan, sehingga pengguna dapat memantau pencapaian dan perkembangan belajarnya secara menyeluruh. Jika menu ini diakses sebelum seluruh materi dan tes akhir selesai, sistem akan memberikan informasi untuk menyelesaikan tahapan tersebut terlebih dahulu.

3.3.4 *Storyboard*

Storyboard yang ditunjukkan pada Tabel 3.2, berfungsi sebagai panduan visual yang merancang setiap adegan (*scene*) dalam aplikasi secara berurutan. Ini membantu memvisualisasikan tampilan antarmuka dan pengalaman pengguna di setiap langkahnya, mulai dari layar pemuatan (*Scene 1*) hingga tampilan sesi belajar berakhir (*Scene 16*).

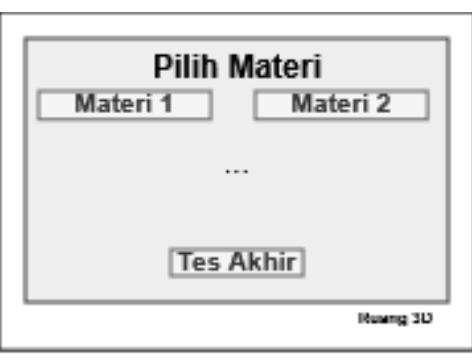
Tabel 3. 2 Storyboard Aplikasi

Scene	Visual	Keterangan
Scene 1		Layar menampilkan <i>loading bar</i> saat aplikasi sedang mempersiapkan semua asset dan konten. Ini memberi tahu pengguna bahwa aplikasi sedang berjalan di latar belakang.
Scene 2		Halaman sampul muncul, menampilkan nama aplikasi dan deskripsi singkat tentang aplikasi. Sebuah tombol yang menonjol mengajak pengguna untuk masuk ke dalam pengalaman.
Scene 3		Aplikasi menyediakan bidang untuk mengetikkan nama, bertujuan agar pengalaman belajar terasa lebih personal. Setelah nama diisi, pengguna dapat menekan tombol untuk ke tahap selanjutnya.
Scene 4		Musik latar yang tenang mulai terdengar saat pengguna masuk ke lingkungan 3D. Avatar 3D bernama Aria muncul untuk memandu pengguna. Ia akan menjelaskan cara navigasi dan kontrol dalam aplikasi. Teks panduan muncul bertahap seiring penjelasan audio dari Aria, sehingga mudah dipahami.

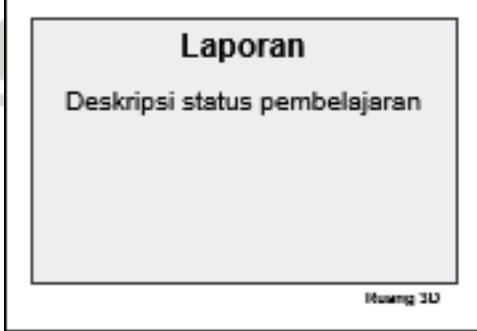
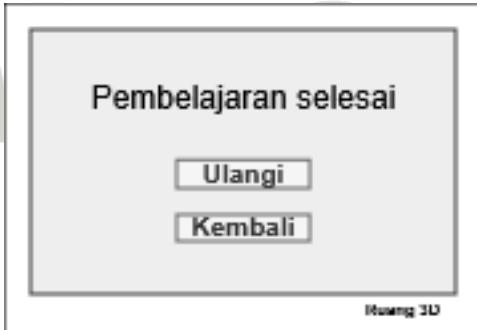
Tabel 3. 3 Storyboard Aplikasi (Lanjutan)

Scene	Visual	Keterangan
Scene 5	 <p>Pilih Mode Pengalaman</p> <p>Desktop</p> <p>VR</p> <p>Ruang 3D</p>	<p>Terdapat panel dengan dua pilihan, yaitu Mode Desktop untuk interaksi di peramban atau Mode VR untuk pengalaman imersif dengan headset.</p>
Scene 6	 <p>Menu Utama</p> <p>Sapaan untuk [user]</p> <p>Belajar</p> <p>Laporan</p> <p>Ruang 3D</p>	<p>Pengguna tiba di menu utama yang menyapa dengan nama mereka. Ada dua pilihan aktivitas untuk langsung eksplorasi materi, dan untuk meninjau skor akhir belajar.</p>
Scene 7	 <p>Pilih Materi</p> <p>Materi 1</p> <p>Terkunci</p> <p>...</p> <p>Terkunci</p> <p>Ruang 3D</p>	<p>Halaman menu pilih materi, menampilkan pilihan materi hanya dapat diakses secara berurutan, sehingga pengguna harus menyelesaikan materi sebelumnya sebelum membuka materi berikutnya, memastikan alur pembelajaran sistematis.</p>
Scene 8	 <p>Deskripsi materi</p> <p>< 1/3 ></p> <p>prev next</p> <p>Ruang 3D</p>	<p>Pengguna berada di halaman pembelajaran dengan model 3D komponen perangkat keras komputer di samping kiri panel informasi. Panel menampilkan deskripsi materi dan dilengkapi tombol audio untuk penjelasan suara serta tombol navigasi untuk mengganti materi. Pengguna dapat melakukan interaksi kontrol kamera dan memutar model 3D.</p>

Tabel 3. 4 Storyboard Aplikasi (Lanjutan)

Scene	Visual	Keterangan
Scene 9		Setelah menyelesaikan materi, sistem menampilkan mini kuis berupa pertanyaan benar/salah untuk menguji pemahaman pengguna. Pengguna harus menjawab kuis ini sebelum dapat melanjutkan ke materi berikutnya.
Scene 10		Sistem menampilkan informasi umpan balik dan tombol di mana akses sesuai hasil jawaban pengguna. Apabila benar maka bisa melanjutkan ke materi selanjutnya, jika salah maka akan mengulang mini kuis atau materi sebelumnya.
Scene 11		Sistem menampilkan informasi bahwa materi telah selesai dipelajari keseluruhan dan dimeriahkan dengan efek <i>konfeti</i> . Pengguna klik tombol untuk melanjutkan ke menu pilih materi sesuai instruksi.
Scene 12		Setelah seluruh materi terbuka, tombol akses tes akhir pada menu materi menjadi aktif. Pengguna dapat memilih untuk mengklik tombol tersebut guna melanjutkan ke tahap tes akhir, atau kembali membaca ulang materi yang sudah terbuka.

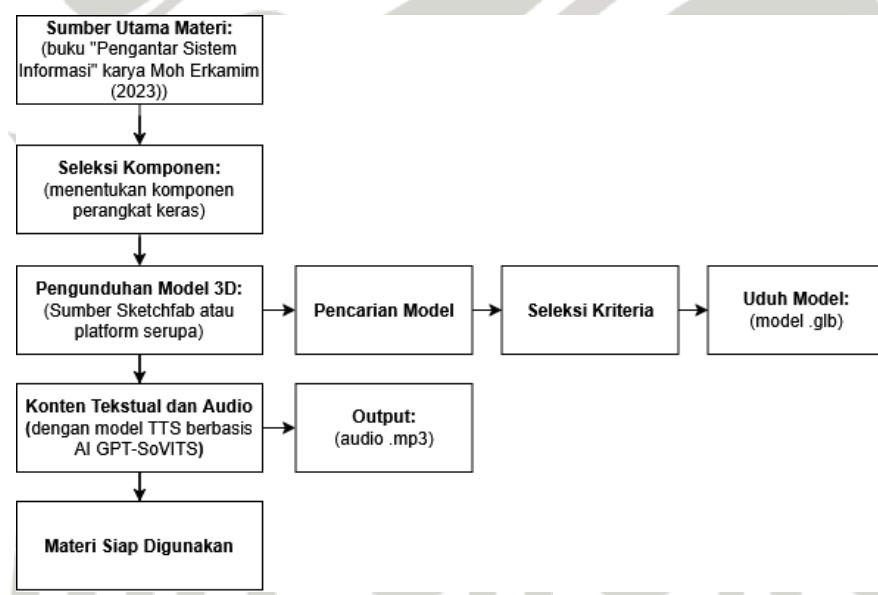
Tabel 3. 5 Storyboard Aplikasi (Lanjutan)

Scene	Visual	Keterangan
Scene 13	 <p>Tes Akhir Soal A B C D Muang 3D</p>	Pengguna mengerjakan Tes Akhir yang berisi pertanyaan dengan empat opsi jawaban. Setiap jawaban diproses oleh sistem untuk menentukan benar atau salah.
Scene 14	 <p>Tes Akhir Umpan balik Tombol Muang 3D</p>	Sistem menampilkan umpan balik status jawaban (benar atau salah) dan menampilkan informasi jawaban yang benar. Pengguna klik tombol untuk pindah ke soal berikutnya.
Scene 15	 <p>Laporan Deskripsi status pembelajaran Muang 3D</p>	Setelah seluruh soal tes akhir selesai dikerjakan, sistem menampilkan ringkasan hasil belajar berupa skor akhir dari tes akhir. Namun apabila halaman ini diakses tanpa mengerjakan tes akhir, maka akan tampil saran penyelesaian materi dan tes akhir terlebih dahulu.
Scene 16	 <p>Pembelajaran selesai Ulangi Kembali Muang 3D</p>	Sesi belajar berakhir, pengguna dapat mengulang materi pembelajaran yang mengarah pada menu pilih materi atau menyelesaikan sesi belajar dengan kembali ke menu utama.

3.4 Material Collection (Pengumpulan Materi)

Tahap pengumpulan materi merupakan bagian penting dalam *Multimedia Development Life Cycle* (MDLC) yang berfungsi sebagai proses pengumpulan dan penyusunan seluruh materi pendukung untuk pembangunan aplikasi. Pada tahap ini, seluruh materi yang dibutuhkan dikumpulkan secara sistematis untuk memenuhi kebutuhan perancangan dan implementasi sistem. Dengan adanya proses ini, pengembangan aplikasi dapat berjalan lebih efisien karena seluruh aset telah tersedia dan siap digunakan.

Struktur proses pengumpulan materi yang digunakan dalam pengembangan aplikasi dapat dilihat pada Gambar 3.8. Diagram tersebut menggambarkan tahapan-tahapan utama dalam pengumpulan dan pengolahan materi pendukung secara sistematis.



Gambar 3. 8 Proses Pengumpulan Materi

Pada tahap pengumpulan materi dalam metodologi MDLC, dilakukan pengumpulan seluruh aset digital yang diperlukan untuk membangun aplikasi. Sumber utama untuk konten materi berasal dari buku "Pengantar Sistem Informasi" karya Moh Erkamim (2023), yang digunakan sebagai dasar untuk melakukan seleksi komponen perangkat keras yang akan ditampilkan.

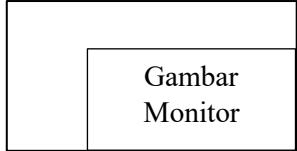
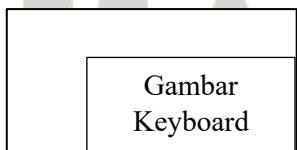
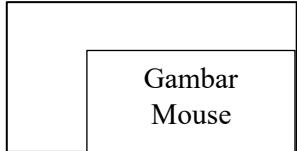
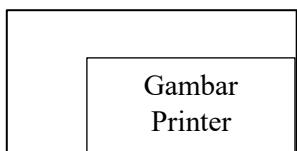
Proses selanjutnya adalah pengumpulan aset visual berupa model 3D, yang dimulai dari pencarian model pada platform seperti Sketchfab. Agar model yang dipilih sesuai dengan kebutuhan aplikasi dan optimal untuk performa, ditetapkan serangkaian kriteria seleksi. Kriteria tersebut dirincikan pada Tabel 3.6 berikut:

Tabel 3. 6 Kriteria Pemilihan Model 3D

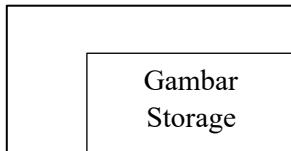
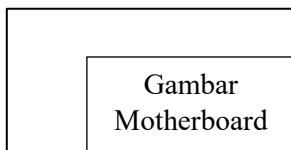
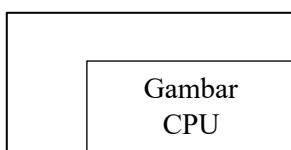
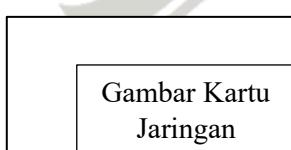
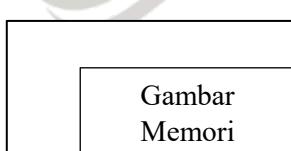
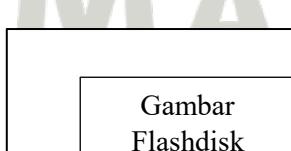
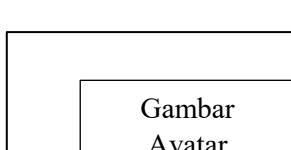
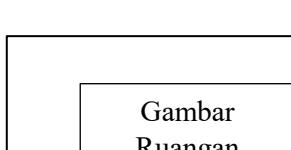
No	Kriteria	Deskripsi
1	Format File	Model harus dalam format GLB (.glb) agar kompatibel dengan Three.js dan WebXR.
2	Detail Visual	Model harus memiliki detail visual yang cukup untuk mengenali komponen dengan jelas namun tidak terlalu kompleks sehingga membebani performa.
3	Lisensi	Model harus memiliki lisensi yang mengizinkan penggunaan untuk edukasi Non Komersial.
4	Kemampuan Optimasi	Model memungkinkan untuk dioptimasi lebih lanjut dengan Blender dan glTF-Transform.

Berdasarkan kriteria tersebut, dilakukan proses pencarian dan pengunduhan model dalam format .glb. Rancangan hasil dari pengumpulan model 3D yang berhasil diperoleh disajikan pada Tabel 3.7.

Tabel 3. 7 Rancangan Hasil Pengumpulan Model 3D

No	Tampilan	Nama Komponen	Sumber
1	 Gambar Monitor	Monitor	Sketchfab/Platform serupa
2	 Gambar Keyboard	Keyboard	Sketchfab/Platform serupa
3	 Gambar Mouse	Mouse	Sketchfab/Platform serupa
4	 Gambar Printer	Printer	Sketchfab/Platform serupa

Tabel 3. 8 Rancangan Hasil Pengumpulan Model 3D (Lanjutan)

No	Tampilan	Nama Komponen	Sumber
5		<i>Storage (HDD/SSD)</i>	Sketchfab/Platform serupa
6		<i>Motherboard</i>	Sketchfab/Platform serupa
7		<i>CPU</i>	Sketchfab/Platform serupa
8		<i>Kartu Jaringan</i>	Sketchfab/Platform serupa
9		<i>Memori</i>	Sketchfab/Platform serupa
10		<i>Kartu Grafis (GPU)</i>	Sketchfab/Platform serupa
11		<i>Flashdisk</i>	Sketchfab/Platform serupa
12		<i>Avatar</i>	Sketchfab/Platform serupa
13		<i>Ruangan</i>	Sketchfab/Platform serupa

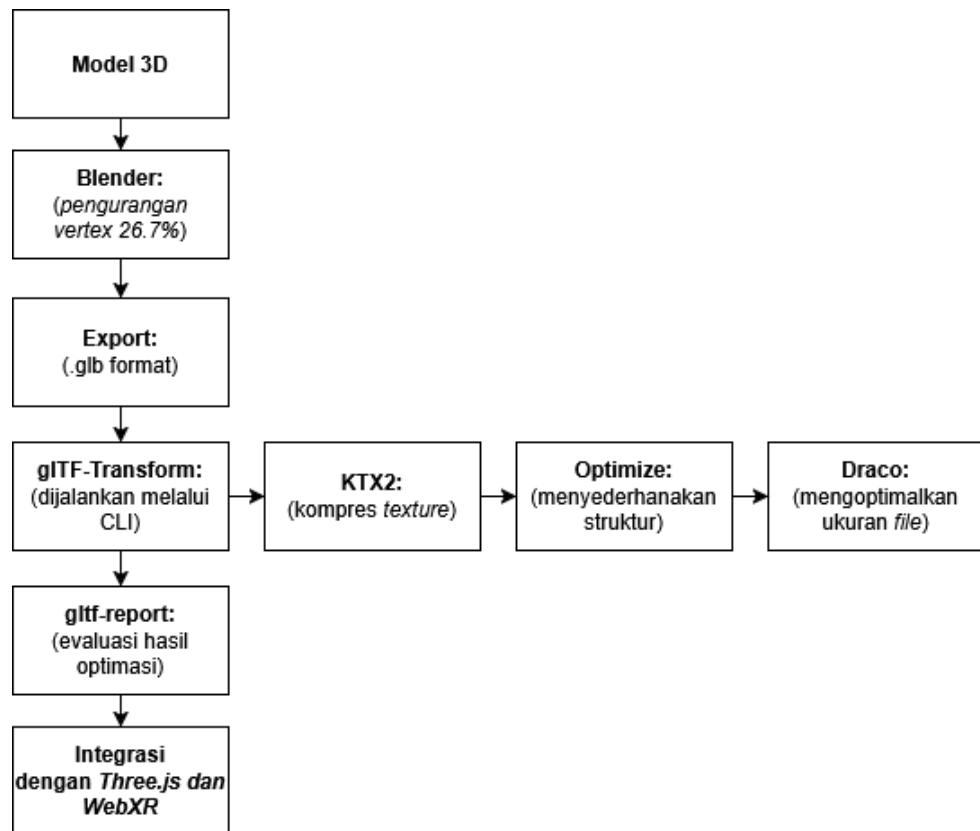
Sementara itu, untuk konten audio, digunakan model GPT-SoVITS sebagai sistem *text-to-speech* (TTS). Model ini telah dilatih sebelumnya menggunakan pasangan data teks dan audio, sehingga pada penelitian ini proses pembuatan suara cukup dilakukan dengan memasukkan teks narasi tanpa perlu menambahkan data pelatihan baru. Hasil konversi dari teks tersebut menghasilkan berkas audio akhir dalam format .mp3. Setelah seluruh model 3D dan konten audio berhasil dihasilkan, materi dianggap siap digunakan pada tahap pengembangan berikutnya.

3.5 *Assembly* (Pembuatan)

Tahap pembuatan ini merupakan implementasi teknis untuk menjawab rumusan masalah yang telah ditetapkan, dengan berfokus pada dua aspek utama yang saling mendukung. Aspek pertama mencakup optimalisasi aset 3D untuk memastikan kinerja aplikasi yang efisien dan ringan. Aspek kedua berkaitan dengan pembangunan arsitektur aplikasi yang mengintegrasikan aset tersebut ke dalam lingkungan Three.js dan WebXR secara fungsional hingga penyajiannya pada perangkat pengguna.

3.5.1 Proses Optimasi Aset 3D

Untuk memastikan pengalaman pengguna yang mulus, terutama dalam mode VR, setiap aset 3D perlu dioptimalkan secara menyeluruh. Alur kerja ini dirancang secara sistematis untuk menekan tiga metrik kinerja utama, yaitu penggunaan VRAM (*Video Random Access Memory*) agar memori grafis lebih efisien, jumlah *draw call* untuk mengurangi beban CPU, serta ukuran *file (disk size)* guna mempercepat waktu muat aplikasi. Proses evaluasi hasil optimasi dilakukan menggunakan glTF-Report, yang membantu menganalisis struktur, ukuran, dan efisiensi aset 3D sebelum diintegrasikan ke dalam sistem.



Gambar 3. 9 Proses Optimasi Model 3D

Alur kerja pertama, yaitu optimasi aset, memastikan setiap model 3D yang digunakan menjadi ringan dan efisien, seperti yang diilustrasikan pada Gambar 3.9. Proses ini diawali dengan optimasi manual di dalam Blender, di mana dilakukan reduksi *vertex* hingga 26,7% untuk menyederhanakan geometri model sebelum diekspor ke format .glb. Selanjutnya, setelah optimasi awal, aset .glb tersebut kemudian dilakukan optimasi lebih lanjut menggunakan alat glTF-Transform melalui serangkaian perintah *Command Line Interface* (CLI). Di sinilah optimasi teknis yang lebih dalam terjadi secara otomatis. Tiga perintah utama dijalankan secara berurutan untuk mencapai efisiensi maksimal:

- 1) Kompresi Tekstur (`etc1s`), perintah ini digunakan untuk mengubah format ke KTX2, sebuah format modern yang dirancang agar data gambar tetap terkompresi bahkan saat sudah berada di VRAM GPU. Langkah ini secara drastis menurunkan penggunaan VRAM, yang sangat vital agar aplikasi tetap lancar di perangkat dengan spesifikasi terbatas.

- 2) Penyederhanaan Struktur (*optimize*), perintah ini bekerja dengan merapikan struktur internal *file*, misalnya dengan menggabungkan beberapa *mesh* yang menggunakan material identik menjadi satu. Tindakan ini secara efektif mengurangi jumlah *draw call*, sehingga meringankan tugas CPU dalam memberi instruksi kepada GPU.
- 3) Kompresi Geometri (*draco*), perintah ini memadatkan data geometri (bentuk *mesh*) menggunakan algoritma kompresi Draco, yang secara signifikan mengurangi ukuran total *file*.

Melalui alur kerja komprehensif ini, dari penyederhanaan manual hingga kompresi otomatis. Setiap aset 3D diubah menjadi versi yang jauh lebih ringan dan efisien, yang siap untuk diintegrasikan ke dalam aplikasi tanpa mengorbankan kinerja. Tabel-tabel berikut merincikan rancangan penurunan dari proses optimasi ini terhadap setiap metrik yang akan diukur.

Tabel 3.9 di bawah ini menjelaskan target pengurangan jumlah *vertex* pada model grafis. Pengurangan ini dilakukan untuk mengurangi beban pemrosesan geometri, sehingga aplikasi dapat berjalan lebih stabil dan responsif, terutama pada perangkat yang memiliki kemampuan grafis atau daya komputasi yang terbatas.

Tabel 3. 9 Rancangan Ringkasan Penurunan *Vertex* pada Model 3D

No	Komponen	Sebelum Optimasi	Sesudah Optimasi
1	Monitor		
2	Keyboard		
3	Motherboard		
4	Mouse		
5	CPU		
6	Memori (RAM)		
7	GPU		
8	Kartu Jaringan		
9	Storage		
10	Printer		
11	Flashdisk		
12	Ruangan		
13	Avatar		
Total			

Kemudian, Tabel 3.10 di bawah ini menjelaskan target pengurangan penggunaan VRAM pada setiap komponen. Pengurangan ini dilakukan untuk menurunkan beban memori grafis sehingga aplikasi dapat berjalan lebih stabil dan responsif, terutama pada perangkat dengan kapasitas VRAM yang terbatas.

Tabel 3. 10 Rancangan Ringkasan Penurunan VRAM pada Model 3D

No	Komponen	Sebelum Optimasi (MB)	Sesudah Optimasi (MB)	Penurunan (%)
1	Monitor			
2	Keyboard			
3	Motherboard			
4	Mouse			
5	CPU			
6	Memori (RAM)			
7	GPU			
8	Kartu Jaringan			
9	Storage			
10	Printer			
11	Flashdisk			
12	Ruangan			
13	Avatar			
Total				

Selanjutnya, Tabel 3.11 menunjukkan rancangan untuk mengurangi jumlah perintah dari CPU ke GPU untuk menggambar objek di layar (*draw call*). Pengurangan ini bertujuan untuk meringankan beban kerja CPU, yang secara langsung berkontribusi pada stabilitas *frame rate*.

Tabel 3. 11 Rancangan Ringkasan Penurunan *Draw Call* pada Model 3D

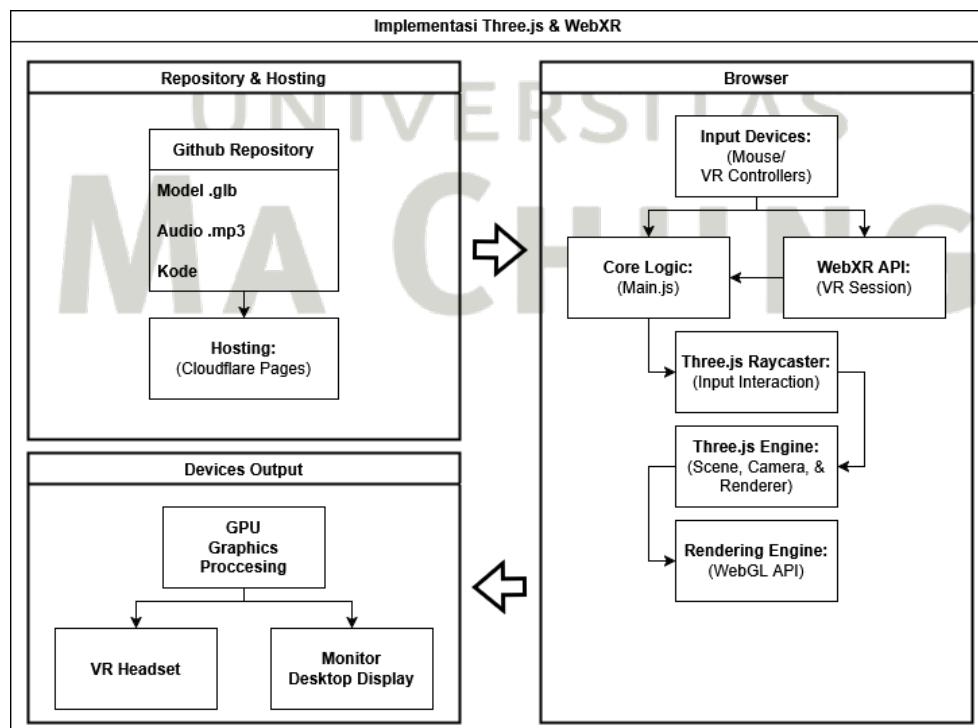
No	Komponen	Sebelum Optimasi	Sesudah Optimasi	Penurunan (%)
1	Monitor			
2	Keyboard			
3	Motherboard			
4	Mouse			
5	CPU			
6	Memori (RAM)			
7	GPU			
8	Kartu Jaringan			
9	Storage			
10	Printer			
11	Flashdisk			
12	Ruangan			
13	Avatar			
Total				

Selanjutnya, Tabel 3.12 memaparkan target optimasi untuk ukuran *disk*. Tujuannya adalah untuk mempercepat waktu muat aplikasi secara signifikan, sehingga meningkatkan pengalaman awal pengguna.

Tabel 3. 12 Rancangan Ringkasan Penurunan Ukuran *Disk* pada Model 3D

No	Komponen	Sebelum Optimasi (MB)	Sesudah Optimasi (MB)	Penurunan (%)
1	Monitor			
2	Keyboard			
3	Motherboard			
4	Mouse			
5	CPU			
6	Memori (RAM)			
7	GPU			
8	Kartu Jaringan			
9	Storage			
10	Printer			
11	Flashdisk			
12	Ruangan			
13	Avatar			
Total				

3.5.2 Arsitektur Aplikasi



Gambar 3. 10 Arsitektur Implementasi Aplikasi Berbasis Three.js dan WebXR

Setelah seluruh asset 3D yang optimal siap, asset tersebut diintegrasikan ke dalam arsitektur teknis aplikasi yang digambarkan pada Gambar 3.10. Arsitektur ini dirancang secara sistematis untuk mengatur alur kerja mulai dari penyimpanan hingga interaksi pengguna. Pada dasarnya, seluruh asset proyek, seperti kode, model .glb, dan audio .mp3, disimpan dan dikelola dalam sebuah repositori GitHub, kemudian didistribusikan kepada pengguna melalui layanan *hosting* Cloudflare Pages. Saat pengguna mengakses aplikasi, seluruh logika inti berjalan di sisi *browser*. *Browser* akan mengelola masukan dari perangkat seperti *mouse* atau *controller VR*, yang kemudian diproses oleh logika utama aplikasi. Logika ini berinteraksi dengan WebXR API untuk menangani sesi VR dan menggunakan Three.js Raycaster untuk mendeteksi interaksi pengguna dengan objek 3D. Selanjutnya, Three.js Engine bertanggung jawab mengelola adegan, kamera, dan proses render dengan memanfaatkan WebGL API untuk menggambar grafis di layar.

Instruksi *rendering* dari peramban ini kemudian diproses oleh GPU (*Graphics Processing Unit*) pada perangkat pengguna, dan hasil visual akhirnya ditampilkan pada monitor *desktop* atau *headset VR*. Untuk mendukung proses pengembangan, ekstensi WebXR API Emulator juga digunakan secara berkala untuk mensimulasikan input VR, sehingga memungkinkan identifikasi dan perbaikan *bug* secara cepat sebelum tahap pengujian formal.

3.6 *Testing (Pengujian)*

Tahap pengujian adalah fase penting dalam MDLC yang bertujuan untuk memvalidasi aplikasi secara sistematis. Tujuannya adalah untuk memastikan bahwa aplikasi yang dibangun telah memenuhi semua kebutuhan yang didefinisikan pada tahap analisis, baik dari segi fungsionalitas, performa, maupun kemudahan penggunaan. Pengujian dilakukan dengan pendekatan yang terstruktur untuk mengidentifikasi dan memperbaiki *bug* atau *error* sebelum aplikasi didistribusikan.

Pada penelitian ini, pengujian dibagi menjadi tiga jenis utama, yaitu pengujian Fungsional (*black box*) untuk memverifikasi fitur, pengujian performa untuk mengukur kinerja teknis, dan pengujian usabilitas (SUS) untuk mengukur pengalaman pengguna.

3.6.1 Pengujian Fungsional (*Black Box*)

Pengujian ini berfokus untuk memverifikasi bahwa setiap fitur aplikasi berjalan sesuai dengan kebutuhan Fungsional yang telah didefinisikan, tanpa memperhatikan struktur kode internalnya. Tujuan utamanya adalah memastikan semua alur interaksi, tombol, dan fungsi bekerja sesuai harapan, baik pada mode *desktop* maupun *headset VR*.

Cara Pelaksanaan Pengujian:

Pengujian dilakukan secara manual oleh peneliti dengan mengikuti skenario yang telah disusun dengan di ulang sebanyak 10x. Peneliti menjalankan aplikasi dan melakukan serangkaian aksi, kemudian membandingkan hasil yang ditampilkan sistem dengan hasil yang diharapkan.

Tabel 3. 13 Rancangan Skenario Pengujian Fungsionalitas *Black Box*

No	Fitur yang Diuji	Skenario Aksi	Hasil yang Diharapkan	Status
1	Alur Awal Aplikasi	Memulai aplikasi, memasukkan nama, dan memilih mode (<i>Browser</i> atau <i>VR</i>).	Aplikasi menampilkan layar selamat datang, menyimpan nama pengguna, dan berhasil masuk ke menu utama sesuai mode yang dipilih.	Belum Diuji
2	Navigasi Menu	Dari menu utama, memilih "Mulai Belajar" dan "Laporan Belajar".	Sistem menampilkan halaman pemilihan materi saat "Mulai Belajar" dipilih dan menampilkan halaman laporan saat "Laporan Belajar" dipilih.	Belum Diuji
3	Tampilan & Interaksi Model	Memilih salah satu materi, melakukan kontrol kamera dan memutar model 3D.	Model 3D komponen yang dipilih ditampilkan dengan benar dan dapat diputar oleh pengguna serta sudut pandang kamera dapat diubah oleh pengguna.	Belum Diuji
4	Informasi & Audio	Menampilkan panel informasi dan menekan tombol audio.	Panel informasi menampilkan teks penjelasan yang sesuai. Audio penjelasan dapat diputar saat tombol ditekan.	Belum Diuji
5	Mini Kuis	Menyelesaikan satu materi dan menjawab mini kuis.	Setelah materi selesai, mini kuis muncul. Setelah dijawab, sistem memberikan umpan balik (benar/salah) dan membuka materi selanjutnya jika jawaban benar.	Belum Diuji

Tabel 3. 14 Rancangan Skenario Pengujian Fungsionalitas *Black Box* (Lanjutan)

No	Fitur yang Diuji	Skenario Aksi	Hasil yang Diharapkan	Status
6	Tes Akhir	Menyelesaikan semua materi dan mengerjakan tes akhir.	Tombol "Tes Akhir" menjadi aktif. Pengguna dapat mengerjakan soal hingga selesai.	Belum Diuji
7	Laporan Belajar	Menyelesaikan tes akhir dan mengakses menu "Laporan Belajar".	Laporan menampilkan skor akhir tes dengan benar.	Belum Diuji
8	Mode VR	Menekan tombol "Mode VR".	Aplikasi berhasil beralih ke mode VR tanpa error, dan interaksi dapat dilakukan menggunakan <i>controller</i> VR.	Belum Diuji
9	Kompatibilitas <i>Browser</i>	Menjalankan aplikasi di <i>browser</i> Chrome, Firefox, dan Edge.	Aplikasi berjalan dengan normal dan fungsionalitas dapat digunakan di ketiga <i>browser</i> tersebut.	Belum Diuji

3.6.2 Pengujian Performa

Pengujian ini bertujuan untuk mengukur dan memastikan aplikasi memenuhi kebutuhan Non Fungsional terkait kinerja. Fokus pengujian adalah pada waktu muat model 3D pada koneksi internet yang telah ditentukan dan kelancaran aplikasi *frame rate* (FPS) pada perangkat *desktop* dan VR. Pengujian dilakukan dengan menggunakan perangkat keras dan perangkat lunak yang ditunjukkan pada Tabel 3.15.

Tabel 3. 15 Spesifikasi Perangkat Pengujian

No	Komponen	Spesifikasi Pengujian
1	Perangkat Laptop	Laptop dengan RAM 16 GB, SSD 512 GB
2	Perangkat VR	Oculus Quest 2
3	<i>Browser</i>	Google Chrome versi 141
4	Jaringan	WiFi 10 Mbps
5	Alat Uji Tambahan	Chrome DevTools dan pustaka <i>stats.module.js</i> .

Cara Pelaksanaan Pengujian:

1) Pengukuran Waktu Muat Model 3D

Pengujian ini akan memanfaatkan Chrome DevTools, khususnya tab Network. Prosedur pengujian dirancang sebagai berikut:

- a. Pengujian ini disimulasikan pada kecepatan jaringan maksimal 10 Mbps menggunakan fitur *Network Monitor* pada Chrome DevTools.
- b. Peneliti akan membersihkan *cache browser* dan melakukan *hard refresh* pada halaman aplikasi.
- c. Waktu muat untuk setiap *file* model 3D (.glb) akan dicatat berdasarkan metrik "Time" pada tab *Network*. Tujuannya adalah untuk memastikan waktu unduh setiap model tidak melebihi 5 detik.

Tujuan dari skenario ini adalah untuk memvalidasi bahwa waktu unduh rata-rata tidak melebihi 5 detik, sehingga waktu tunggu pada saat pemuatan sistem tidak terlalu lama. Hasil pengujian akan dicatat pada Tabel 3.16.

Tabel 3. 16 Rancangan Uji Performa Pengukuran Waktu Muat Model 3D

No	Komponen	Waktu Muat (detik)
1	Monitor	
2	Keyboard	
3	Mouse	
4	Printer	
5	Storage (HDD/SSD)	
6	Motherboard	
7	CPU	
8	Kartu Jaringan	
9	Memori (RAM)	
10	Kartu Grafis (GPU)	
11	Flashdisk	
12	Avatar 3D	
13	Ruangan	
Rata-Rata		

2) Pengukuran *Frame Rate* (FPS)

Untuk mengukur kelancaran visualisasi, pustaka JavaScript *stats.module.js* akan diintegrasikan ke dalam aplikasi. Pustaka ini akan menampilkan overlay penghitung FPS secara *real-time* di layar. Peneliti akan melakukan serangkaian skenario interaksi yang telah didefinisikan, baik pada mode *desktop* maupun VR. Nilai FPS akan diamati selama skenario tersebut berlangsung untuk mengidentifikasi kinerja rata-rata dan potensi penurunan performa. Skenario pengujian dirinci pada Tabel 3.17.

Tabel 3. 17 Rancangan Uji Performa Pengukuran FPS

No	Mode Pengujian	Skenario Uji	Hasil Observasi (FPS)	Status
1	Mode Desktop	Interaksi normal dengan navigasi UI, rotasi kamera, serta pemutaran model 3D selama 60 detik.		Belum Diuji
2	Mode Desktop	Uji beban dilakukan saat memuat halaman tutorial, yang memiliki model avatar dengan tingkat kompleksitas tertinggi.		Belum Diuji
3	Mode VR	Interaksi normal dengan navigasi UI dan rotasi model menggunakan <i>controller</i> VR selama 60 detik.		Belum Diuji
4	Mode VR	Uji beban saat memuat halaman tutorial dalam mode VR.		Belum Diuji

3.6.3 Pengujian Kegunaan dengan *System Usability Scale* (SUS)

Pengujian ini bertujuan untuk mengukur tingkat kemudahan penggunaan dan kepuasan pengguna terhadap aplikasi secara subjektif, dengan fokus pada perolehan skor kuantitatif mengenai persepsi usabilitas, identifikasi sejauh mana antarmuka aplikasi intuitif dan mudah dipelajari oleh target pengguna.

Cara Pelaksanaan Pengujian:

- 1) Melibatkan sejumlah 30 partisipan yang sesuai dengan persona pengguna utama, yaitu mahasiswa/peserta didik dan masyarakat umum dengan minat teknologi.
- 2) Setiap partisipan diberikan pengarahan singkat mengenai tujuan aplikasi.
- 3) Partisipan diminta untuk mencoba aplikasi dan menyelesaikan keseluruhan materi hingga tes akhir.
- 4) Setelah selesai, partisipan diminta untuk mengisi kuesioner SUS yang terdiri dari 10 pernyataan (Tabel 3.18), dengan skala Likert 5 poin, mulai dari 1 sangat tidak setuju, 2 tidak setuju, 3 netral, 4 setuju, dan 5 sangat setuju.
- 5) Skor dari semua partisipan dikumpulkan dan dihitung menggunakan rumus standar SUS untuk mendapatkan skor akhir (rentang 0-100).

Tabel 3. 18 Format Kuesioner SUS

No	Pernyataan	Skor (1–5)
1	Saya berpikir akan sering menggunakan aplikasi ini.	
2	Saya merasa aplikasi ini rumit untuk digunakan.	
3	Saya merasa aplikasi ini mudah dipelajari.	
4	Saya memerlukan bantuan orang lain atau dukungan teknis untuk menggunakan aplikasi ini.	
5	Saya merasa berbagai fungsi dalam aplikasi ini terintegrasi dengan baik.	
6	Saya merasa ada terlalu banyak inkonsistensi dalam aplikasi ini	
7	Saya merasa banyak orang akan belajar menggunakan aplikasi ini dengan cepat.	
8	Saya merasa aplikasi ini membungkung saat digunakan.	
9	Saya merasa percaya diri saat menggunakan aplikasi ini.	
10	Saya harus mempelajari banyak hal terlebih dahulu sebelum bisa menggunakan aplikasi ini.	

Interpretasi skor SUS:

- 1) Skor > 80 : usabilitas sangat baik.
- 2) Skor ≥ 68 : usabilitas dapat diterima.
- 3) Skor < 68 : perlu perbaikan signifikan.

3.7 **Distribution (Distribusi)**

Tahap akhir ini bertujuan untuk mendistribusikan aplikasi agar dapat diakses oleh pengguna. Distribusi dilakukan melalui Cloudflare Pages, layanan *hosting* gratis yang terintegrasi dengan repositori proyek. Proses ini menggunakan pendekatan *Continuous Deployment*, di mana setiap pembaruan kode akan secara otomatis memicu proses *build* dan *deploy* versi terbaru. Rincian distribusi disajikan pada Tabel 3.19.

Tabel 3. 19 Rincian Distribusi Aplikasi

No	Komponen	Deskripsi
1	Platform Hosting	Cloudflare Pages
2	Repository Source	GitHub
3	Protokol Akses	HTTPS
4	Metode Deploy	Continuous Deployment via GitHub Actions
5	Link Akses	URL publik dari Cloudflare Pages

Bab IV

Hasil dan Pembahasan

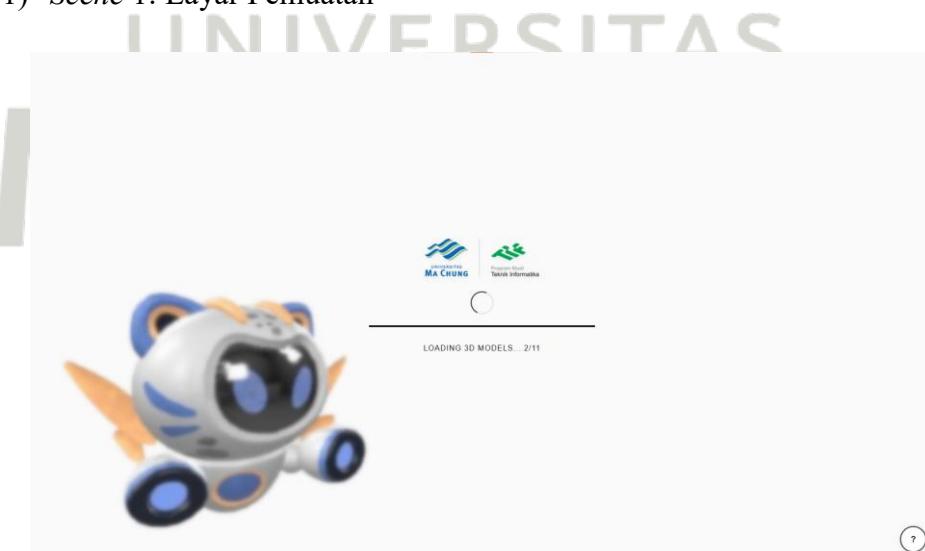
4.1 Hasil Konsep dan Perancangan

Bagian ini merupakan hasil akhir dari keseluruhan proses perancangan yang telah dijelaskan pada Bab III. Subbab ini menyajikan hasil implementasi secara terpadu sebagai representasi dari rancangan yang telah dibuat. Penjelasan disusun berdasarkan alur visual yang mengikuti urutan adegan demi adegan (*scene by scene*) sesuai dengan rancangan *storyboard*. Pendekatan ini dipilih karena setiap adegan yang ditampilkan secara langsung merepresentasikan alur fungsional yang telah dirancang melalui *flowchart* dan *activity diagram*. Dengan demikian, penyajian ini sekaligus memastikan bahwa seluruh kebutuhan fungsional yang telah ditetapkan pada tahap konsep dapat terpenuhi.

4.1.1 *Scene 1-6: Mengelola Sesi*

Pada tahap awal penggunaan aplikasi, pengguna akan melalui serangkaian proses persiapan sesi. Tahapan ini mencakup proses pemuatan aset, personalisasi pengguna, tutorial penggunaan, pemilihan mode interaksi, hingga akses ke menu utama.

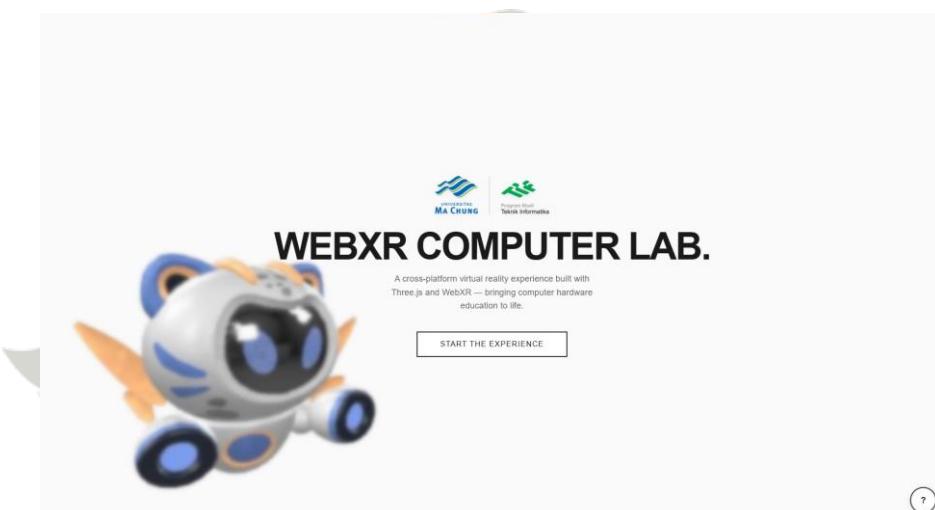
- 1) *Scene 1: Layar Pemuatan*



Gambar 4. 1 Tampilan Layar Pemuatan

Sesuai dengan perancangan, aplikasi diawali dengan tampilan layar pemuatan (*loading screen*) seperti yang terlihat pada (Gambar 4.1). Halaman ini berfungsi memberikan indikator visual kepada pengguna bahwa sistem sedang mengunduh dan mempersiapkan seluruh aset digital, termasuk model 3D, tekstur, dan audio. Indikator progres ditampilkan untuk memastikan pengguna mengetahui bahwa aplikasi sedang berproses dan tidak mengalami gangguan (*freeze*) saat memuat data yang cukup besar.

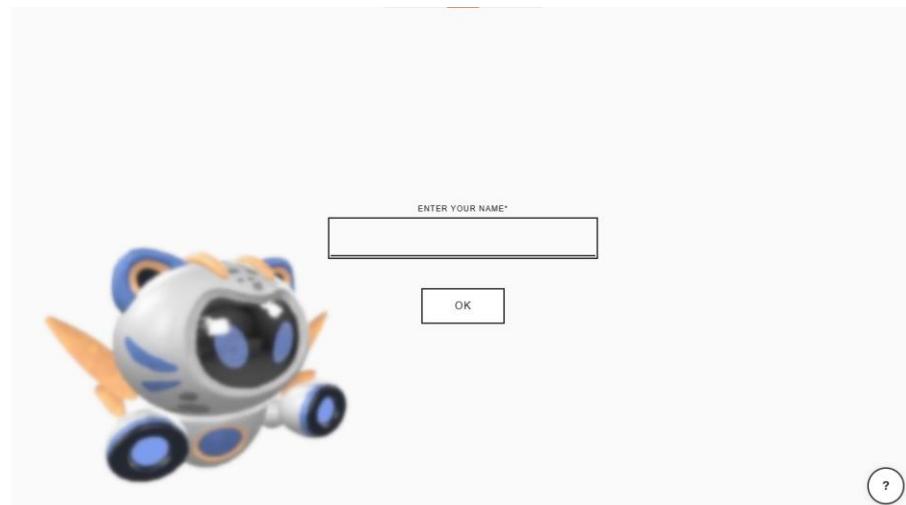
2) *Scene 2: Halaman Sampul*



Gambar 4. 2 Tampilan Halaman Sampul

Setelah proses pemuatan selesai, sistem menampilkan halaman sampul (Gambar 4.2). Tampilan ini berfungsi sebagai gerbang utama aplikasi yang menyajikan judul dan deskripsi singkat mengenai tujuan aplikasi. Antarmuka dirancang sederhana dengan satu tombol aksi utama untuk memulai pengalaman, meminimalisir kebingungan pengguna saat pertama kali mengakses aplikasi.

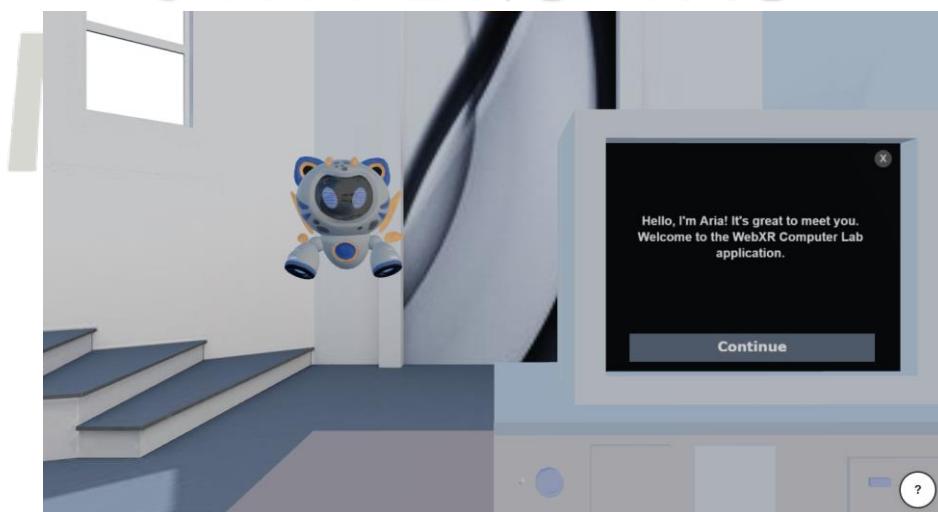
3) *Scene 3: Halaman Input Nama*



Gambar 4. 3 Tampilan Halaman Input Nama

Sebelum masuk ke materi, aplikasi menampilkan halaman input nama (Gambar 4.3) untuk memenuhi kebutuhan personalisasi. Pada halaman ini, pengguna diminta memasukkan nama panggilan. Sistem akan menyimpan data input ini sementara untuk digunakan dalam menyapa pengguna di berbagai bagian aplikasi, sehingga menciptakan pengalaman belajar yang lebih personal dan menarik.

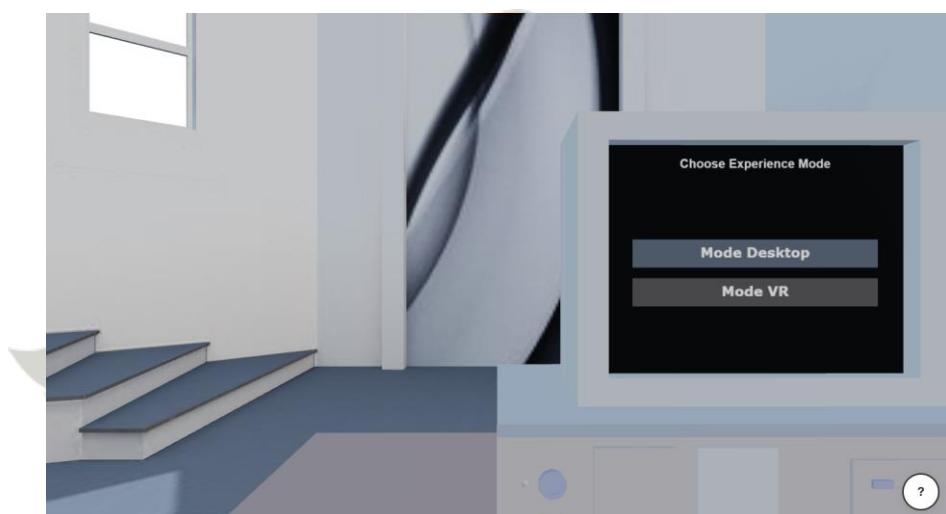
4) *Scene 4: Halaman Tutorial*



Gambar 4. 4 Tampilan Halaman Tutorial

Setelah data pengguna tersimpan, pengguna diarahkan masuk ke dalam lingkungan 3D untuk pertama kalinya (Gambar 4.4). Pada tahap ini, sistem menampilkan karakter avatar 3D pemandu bernama "Aria". Fungsi utama adegan ini adalah memberikan tutorial singkat mengenai cara navigasi dan interaksi. Penjelasan disajikan melalui kombinasi teks yang muncul secara bertahap dan narasi audio, memastikan pengguna memahami cara mengoperasikan aplikasi sebelum memulai sesi belajar.

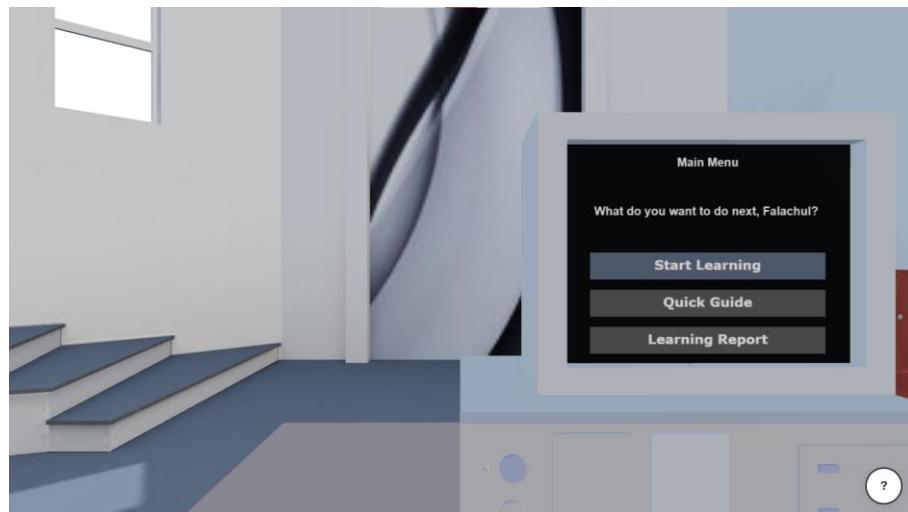
5) *scene 5*: Halaman Pilih Mode



Gambar 4. 5 Tampilan Halaman Pilih Mode

Aplikasi ini dirancang untuk mendukung lintas platform, sehingga sistem menyediakan halaman pilih mode (Gambar 4.5). Pengguna diberikan opsi untuk memilih mode operasi sesuai perangkat yang digunakan, yaitu mode *desktop* untuk penggunaan standar menggunakan *mouse*, atau mode VR untuk pengalaman imersif menggunakan *headset* VR. Pilihan ini akan menentukan mekanisme kontrol yang diaktifkan oleh sistem selanjutnya.

6) Scene 6: Halaman Menu Utama



Gambar 4. 6 Tampilan Halaman Menu Utama

Setelah memilih mode dan menyelesaikan tutorial, pengguna diarahkan ke menu utama (Gambar 4.6). Halaman ini menjadi pusat navigasi aplikasi. Sesuai dengan fitur personalisasi, sistem menyapa pengguna menggunakan nama yang telah diinputkan sebelumnya. Di sini, pengguna disajikan opsi utama sesuai rancangan alur kerja, yaitu opsi untuk mulai belajar guna masuk ke materi, atau laporan belajar untuk melihat riwayat skor evaluasi.

4.1.2 Scene 7-11: Sesi Pembelajaran

Bagian ini merupakan inti dari aplikasi, di mana pengguna berinteraksi langsung dengan materi pengenalan perangkat keras komputer. Alur ini mencakup pemilihan topik, eksplorasi model 3D, hingga evaluasi ringan melalui mini kuis.

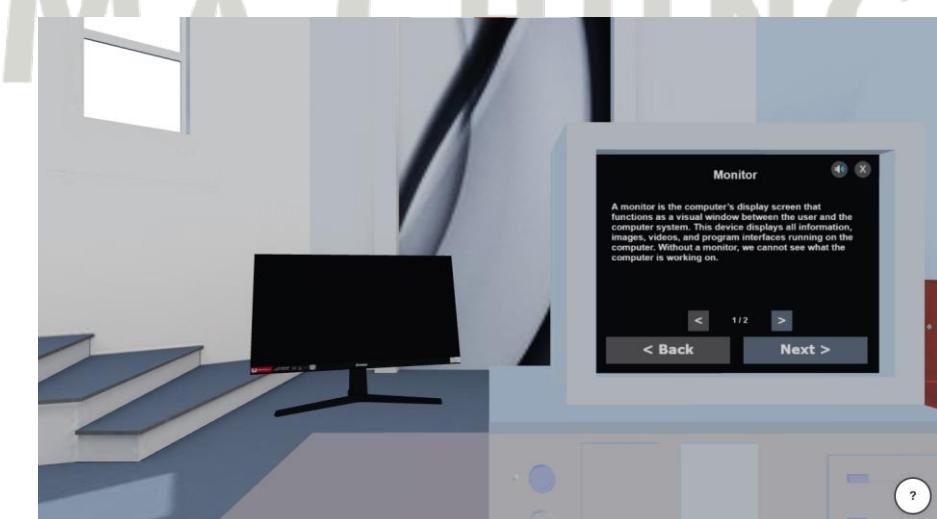
1) *Scene 7: Halaman Pilih Materi*



Gambar 4. 7 Halaman Pilih Materi

Saat pengguna memilih untuk mulai belajar, sistem menampilkan halaman pilih materi (Gambar 4.7). Antarmuka ini menyajikan daftar topik perangkat keras yang tersedia. Untuk memastikan proses belajar yang terstruktur dan sistematis, sistem menerapkan logika penguncian materi. Materi selanjutnya hanya dapat diakses jika materi sebelumnya telah diselesaikan. Selain itu, tombol akses menuju tes akhir juga ditampilkan namun dalam kondisi Non Aktif, dan baru akan terbuka setelah seluruh materi selesai dipelajari.

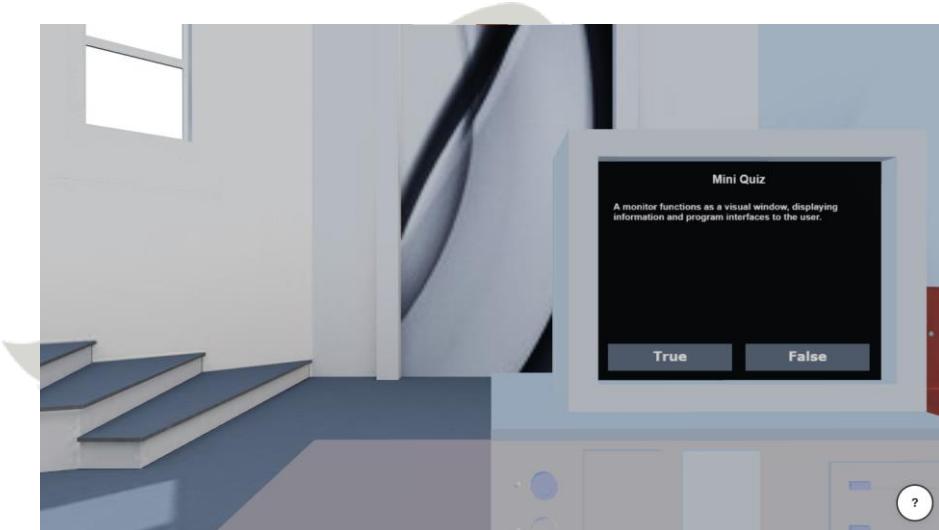
2) *Scene 8: Halaman Pembelajaran Materi*



Gambar 4. 8 Halaman Pembelajaran Materi

Tampilan ini (Gambar 4.8) merealisasikan kebutuhan utama aplikasi sebagai media pengenalan perangkat keras. Layar terbagi menjadi dua area fokus, yaitu area visualisasi yang menampilkan model 3D komponen secara interaktif, dan area panel informasi yang memuat deskripsi materi. Pengguna dapat memutar model 3D untuk melihat detail komponen dari berbagai sisi. Panel informasi juga dilengkapi dengan tombol kontrol audio untuk memutar penjelasan suara, serta tombol navigasi untuk berpindah antar halaman materi.

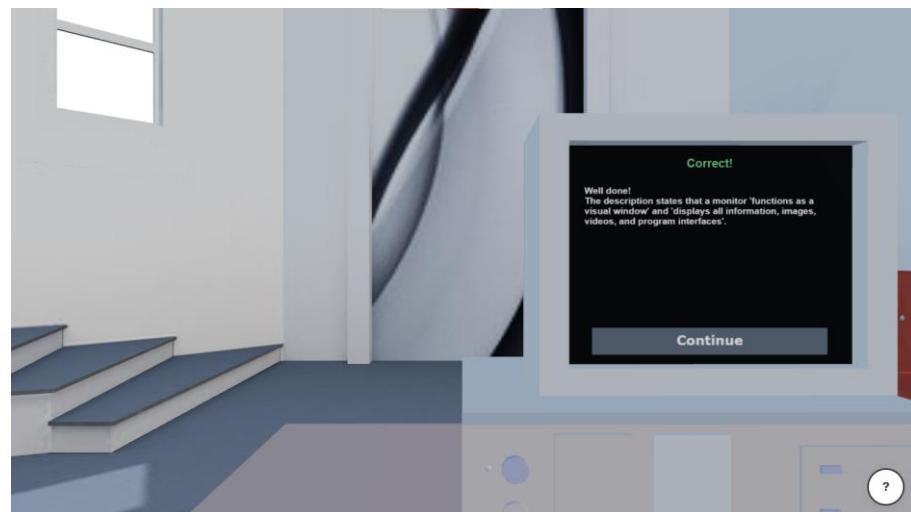
3) *Scene 9: Halaman Mini Kuis*



Gambar 4. 9 Halaman Mini Kuis

Setelah pengguna menyelesaikan membaca atau mendengarkan materi pada satu topik, sistem akan mengarahkan pengguna ke halaman mini kuis (Gambar 4.9). Fitur ini berfungsi sebagai evaluasi formatif untuk menguji pemahaman pengguna secara instan. Antarmuka menampilkan satu pertanyaan relevan dengan materi yang baru saja dipelajari, disertai opsi jawaban benar atau salah.

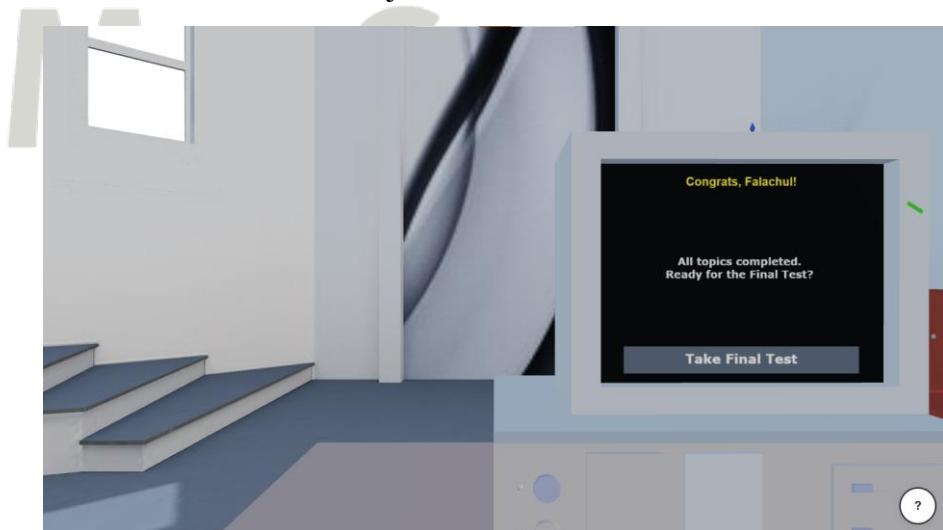
4) *Scene 10: Halaman Umpan Balik Mini Kuis*



Gambar 4. 10 Hasil Halaman Umpan Balik Mini Kuis

Segera setelah pengguna memilih jawaban pada mini kuis, sistem memberikan respons langsung melalui halaman umpan balik (Gambar 4.10). Jika jawaban pengguna benar, sistem akan menampilkan validasi positif dan tombol untuk melanjutkan ke materi berikutnya. Sebaliknya, jika jawaban salah, sistem memberikan informasi koreksi dan opsi untuk mengulang materi. Mekanisme ini memastikan pengguna benar-benar memahami materi sebelum melangkah ke topik selanjutnya.

5) *Scene 11: Halaman Sesi Belajar Selesai*



Gambar 4. 11 Hasil Halaman Sesi Belajar Selesai

Tampilan ini (Gambar 4.11) muncul ketika pengguna telah berhasil menyelesaikan seluruh rangkaian materi dan mini kuis. Sistem memberikan apresiasi visual berupa efek konfetti dan ucapan selamat untuk meningkatkan kepuasan pengguna. Pada tahap ini, sistem menginformasikan bahwa seluruh materi telah tuntas, dan mengarahkan pengguna kembali ke menu pemilihan materi di mana akses menuju tes akhir kini telah terbuka.

4.1.3 *Scene 12-16: Evaluasi Akhir*

Alur ini merupakan implementasi dari *use case* mengerjakan tes akhir dan melihat laporan belajar. Tahapan ini dimulai setelah pengguna menyelesaikan semua materi pembelajaran (*scene 11*) dan kembali ke menu materi, di mana fungsionalitas tes akhir kini telah terbuka.

1) *Scene 12: Halaman Pilih Materi Akses Final Test*



Gambar 4. 12 Halaman Pilih Materi (Akses Tes Akhir)

Pengguna diarahkan kembali ke halaman pilih materi (Gambar 4.12) dengan status materi yang telah diperbarui. Karena seluruh topik telah diselesaikan, tombol tes akhir yang sebelumnya terkunci kini menjadi aktif. Perubahan status visual ini memberikan petunjuk yang jelas kepada pengguna mengenai langkah selanjutnya yang harus diambil.

2) *Scene 13: Halaman Soal Final Test*



Gambar 4. 13 Hasil Halaman Soal Final Test

Pada sesi evaluasi ini, sistem menyajikan halaman soal tes akhir (Gambar 4.13) dalam format pilihan ganda (*multiple choice*). Berbeda dengan mini kuis, tes ini mencakup pertanyaan dari seluruh materi yang telah dipelajari. Antarmuka menampilkan pertanyaan, opsi jawaban, serta indikator nomor soal untuk membantu pengguna mengetahui progres penggerjaan tes.

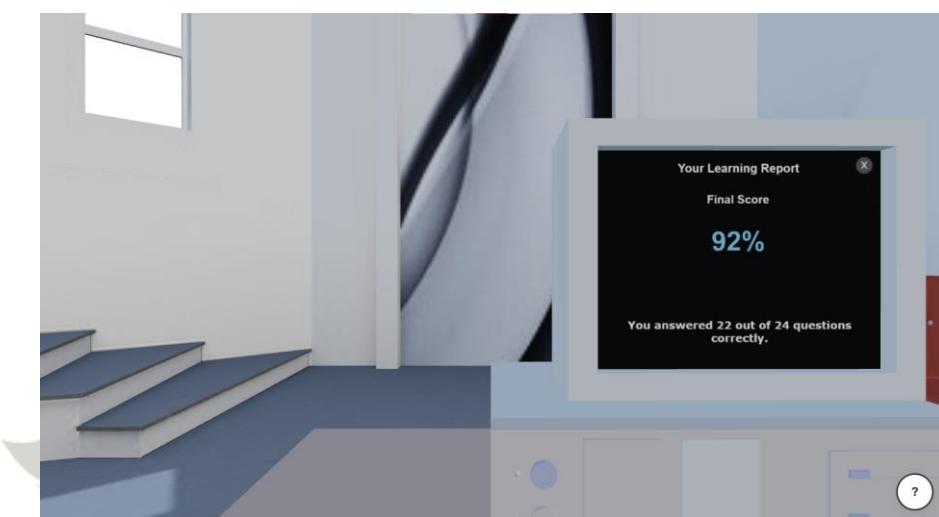
3) *Scene 14: Halaman Umpam Balik Jawaban Final Test*



Gambar 4. 14 Hasil Halaman Umpam Balik Jawaban Final Test

Setiap kali pengguna menjawab satu soal dalam tes akhir, sistem menampilkan umpan balik jawaban (Gambar 4.14). Halaman ini menginformasikan apakah jawaban yang dipilih benar atau salah, serta menyertakan kunci jawaban yang tepat sebagai penguatan pemahaman. Pengguna kemudian dapat menekan tombol untuk melanjutkan ke butir soal berikutnya hingga seluruh soal terjawab.

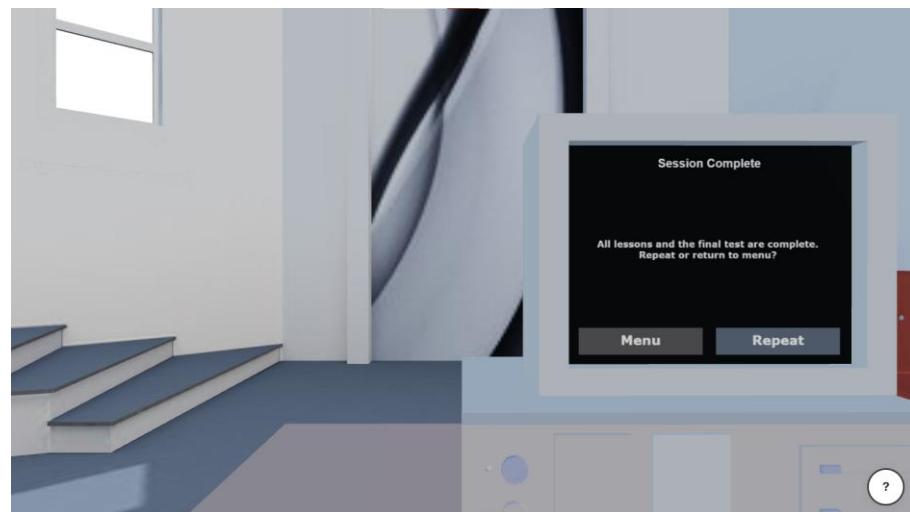
4) Scene 15: Halaman Laporan Akhir



Gambar 4. 15 Hasil Halaman Laporan Akhir

Setelah seluruh soal tes akhir diselesaikan, sistem memproses hasil jawaban dan menampilkannya pada halaman laporan akhir (Gambar 4.15). Halaman ini menyajikan ringkasan performa belajar pengguna berupa skor akhir dalam format persentase serta jumlah jawaban yang benar. Informasi ini memberikan gambaran objektif mengenai tingkat keberhasilan pengguna dalam memahami materi perangkat keras komputer.

5) *Scene 16: Halaman Sesi Akhir*



Gambar 4. 16 Hasil Halaman Sesi Belajar Berakhir

Sebagai penutup alur penggunaan aplikasi, sistem menampilkan halaman sesi akhir (Gambar 4.16). Antarmuka ini memberikan opsi navigasi akhir kepada pengguna, yaitu pilihan untuk kembali ke menu utama guna melihat laporan ulang, atau kembali ke daftar materi jika pengguna ingin mengulang pembelajaran tertentu.

4.2 Hasil Pengumpulan Materi

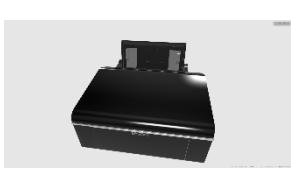
Sub-bab ini menguraikan realisasi dari tahap Pengumpulan Materi sebagaimana telah dirancang dalam alur penelitian pada Bab III. Fokus utama pada tahap ini adalah penyediaan dua jenis aset digital yang menjadi fondasi aplikasi, yaitu naskah narasi untuk kebutuhan audio serta model 3D sebagai representasi visual komponen.

Terkait konten audio, naskah narasi disusun untuk menjelaskan setiap komponen perangkat keras. Penulisan naskah dilakukan menggunakan Bahasa Inggris guna memenuhi spesifikasi teknis model sintesis suara berbasis AI, GPT-SOVITS, yang memproses teks masukan menjadi keluaran audio berformat .mp3. Materi narasi yang dikumpulkan mencakup 12 topik pembahasan, meliputi *Introduction, Monitor, Keyboard, Mouse, Motherboard, Processor (CPU), Memory (RAM), Graphics Card (GPU), Network Card (NIC), Storage, Printer, dan Flash*

Drive. Naskah narasi lengkap untuk seluruh materi tersebut dapat dilihat secara rinci pada Lampiran A.

Sementara itu, pengumpulan aset visual dilakukan dengan mengakuisisi model 3D dari platform daring seperti Sketchfab dan CGTrader. Proses ini mengacu pada kriteria seleksi yang telah ditetapkan sebelumnya, seperti penggunaan format .glb, kesesuaian lisensi, serta detail visual yang memadai. Sebanyak 13 model 3D berhasil dikumpulkan untuk diintegrasikan ke dalam aplikasi, sebagaimana dirincikan pada Tabel 4.1 berikut.

Tabel 4. 1 Hasil Pengumpulan Model 3D

No	Tampilan	Nama Komponen	Sumber
1		Monitor	Sketchfab
2		Keyboard	CGTrader
3		Mouse	Sketchfab
4		Printer	Sketchfab
5		Storage (HDD/SSD)	Sketchfab

Tabel 4. 2 Hasil Pengumpulan Model 3D (Lanjutan)

No	Tampilan	Nama Komponen	Sumber
6		<i>Motherboard</i>	Sketchfab
7		CPU	Sketchfab
8		Kartu Jaringan	Sketchfab
9		Memori	Sketchfab
10		Kartu Grafis (GPU)	Sketchfab
11		Flashdisk	Sketchfab
12		Avatar	Sketchfab
13		Ruangan	Sketchfab

4.3 Hasil Pembuatan

Bagian ini merealisasikan rancangan teknis pada Bab III menjadi aplikasi fungsional. Subbab ini memaparkan hasil tahap pembuatan (*Assembly*), yang berfokus pada optimasi aset 3D agar aplikasi efisien dan ringan, serta arsitektur aplikasi yang mengintegrasikan dan menjalankan aset-aset tersebut dalam lingkungan Three.js dan WebXR.

4.3.1 Hasil Optimasi Model 3D

Tahap ini menerapkan alur optimasi Bab III pada seluruh aset 3D, dimulai dengan reduksi *vertex* 26,7% menggunakan *Decimate* di Blender. Model .glb kemudian diproses dengan glTF-Transform melalui CLI untuk kompresi tekstur ke KTX2, penyederhanaan *mesh*, dan kompresi geometri dengan Draco, sehingga ukuran *file* berkurang signifikan. Efektivitas optimasi dievaluasi menggunakan glTF-Report berdasarkan penurunan *vertex*, penggunaan VRAM, jumlah *draw call*, dan ukuran *disk*.

1) Penurunan *Vertex* pada Model 3D

Optimasi geometri dilakukan menggunakan teknik *Decimate* di Blender untuk menyederhanakan struktur model 3D. Proses ini berhasil mengurangi total *vertex* dari 605.652 menjadi 443.903. Pengurangan ini secara langsung meringankan beban komputasi geometri pada GPU tanpa mengurangi kualitas visual secara signifikan.

Tabel 4. 3 Ringkasan Penurunan *Vertex* pada Model 3D

No	Komponen	Sebelum Optimasi	Sesudah Optimasi
1	Monitor	6.564	4.811
2	Keyboard	261.509	191.698
3	Motherboard	84.036	61.586
4	Mouse	17.688	12.957
5	CPU	64.682	47.406
6	Memori (RAM)	1.380	1.011
7	GPU	17.304	12.681
8	Kartu Jaringan	129.205	94.689
9	Storage	4.550	3.335
10	Printer	4.620	3.385
11	Flashdisk	1.062	778
12	Ruangan	10.395	7.619
13	Avatar	2.657	1.947
Total		605.652	443.903

2) Penurunan VRAM pada Model 3D

Kompresi tekstur KTX2 memberikan kontribusi terbesar dalam menurunkan jejak VRAM karena tekstur tetap terkompresi saat diproses GPU. Total penggunaan VRAM seluruh model 3D turun 80,40%, dari 238,8 MB menjadi 46,785 MB, memastikan aplikasi tetap optimal pada perangkat dengan VRAM terbatas.

Tabel 4. 4 Ringkasan Penurunan VRAM pada Model 3D

No	Komponen	Sebelum Optimasi (MB)	Sesudah Optimasi (MB)	Penurunan (%)
1	Monitor	20,4	2,6	87,25%
2	Keyboard	14,9	1,4	90,60%
3	Motherboard	58,3	8,2	85,93%
4	Mouse	17,5	2,2	87,43%
5	CPU	20,4	2,9	85,78%
6	Memori (RAM)	30,8	3,9	87,34%
7	GPU	28,8	3,8	86,81%
8	Kartu Jaringan	6,5	0,866	86,68%
9	Storage	0,191	0,064	66,49%
10	Printer	0,195	0,046	76,41%
11	Flashdisk	0,123	0,049	60,16%
12	Ruangan	17,2	16,9	1,74%
13	Avatar	22,5	2,9	87,11%
Total		238,8	46,785	80,40%

3) Penurunan Draw Call pada Model 3D

Penyederhanaan struktur model melalui perintah *optimize* pada glTF-Transform bertujuan untuk mengurangi jumlah instruksi *render* (*draw call*) dari CPU ke GPU. Dengan menggabungkan *mesh* yang memiliki material serupa, beban CPU dapat ditekan. Proses ini berhasil mengurangi total *draw call* sebesar 71,21%, dari 198 menjadi 57 instruksi. Pengurangan ini secara langsung meningkatkan efisiensi *rendering* dan menjaga *frame rate* tetap stabil.

Tabel 4. 5 Ringkasan Penurunan *Draw Call* pada Model 3D

No	Komponen	Sebelum Optimasi	Sesudah Optimasi	Penurunan (%)
1	Monitor	4	4	0,00%
2	Keyboard	13	3	76,92%
3	Motherboard	36	9	75,00%
4	Mouse	7	5	28,57%
5	CPU	1	1	0,00%
6	Memori (RAM)	4	4	0,00%
7	GPU	11	8	27,27%
8	Kartu Jaringan	82	3	96,34%
9	Storage	3	1	66,67%
10	Printer	5	5	0,00%
11	Flashdisk	9	1	88,89%
12	Ruangan	14	4	71,43%
13	Avatar	9	9	0,00%
Total		198	57	71,21%

4) Penurunan Ukuran *Disk* pada Model 3D

Kombinasi dari reduksi *vertex*, kompresi geometri Draco, dan kompresi tekstur KTX2 secara drastis mengurangi ukuran *file* asset. Ukuran *file* yang lebih kecil mempercepat waktu muat aplikasi secara signifikan. Secara total, ukuran seluruh aset 3D berhasil dikurangi sebesar 84,97%, dari 40,03 MB menjadi 6,013 MB, membuat aplikasi lebih cepat diakses oleh pengguna.

Tabel 4. 6 Ringkasan Penurunan Ukuran *Disk* pada Model 3D

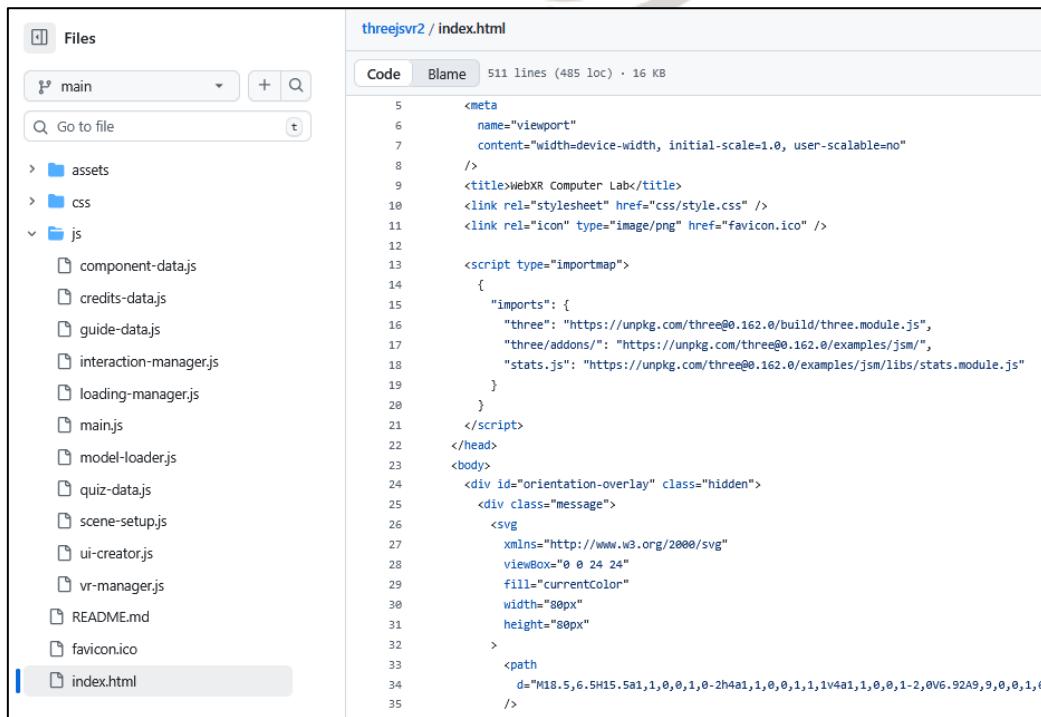
No	Komponen	Sebelum Optimasi (MB)	Sesudah Optimasi (MB)	Penurunan (%)
1	Monitor	2,5	0,35	86,00%
2	Keyboard	14,9	1,4	90,60%
3	Motherboard	6,9	1,8	73,91%
4	Mouse	1,1	0,16	85,45%
5	CPU	3,8	0,89	76,58%
6	Memori (RAM)	0,96	0,15	84,38%
7	GPU	1,3	0,38	70,77%
8	Kartu Jaringan	6,5	0,799	87,71%
9	Storage	0,193	0,065	66,32%
10	Printer	0,201	0,050	75,12%
11	Flashdisk	0,131	0,049	62,60%
12	Ruangan	0,735	0,297	59,59%
13	Avatar	1,4	0,263	81,21%
Total		40,03	6,013	84,97%

4.3.2 Hasil Arsitektur Aplikasi

Sub-bab ini memaparkan hasil implementasi teknis dari arsitektur aplikasi yang telah dirancang pada Gambar 3.10. Implementasi ini mencakup alur kerja lengkap aplikasi, mulai dari manajemen repositori dan *hosting* di sisi *server*, dilanjutkan dengan inisialisasi *engine* dan logika interaksi di sisi *browser*, hingga proses *rendering* akhir ke perangkat *output* pengguna.

1) Manajemen Repository dan Distribusi (*Hosting*)

Sesuai rancangan arsitektur, seluruh kode sumber (JavaScript, HTML, CSS) dan aset aplikasi (model .glb, audio) dikelola menggunakan repositori Git pada platform GitHub. Untuk distribusi aplikasi agar dapat diakses publik, penelitian ini memanfaatkan layanan *hosting* Cloudflare Pages. Layanan ini dihubungkan langsung ke repositori GitHub, sehingga setiap perubahan pada kode sumber dapat secara otomatis terpublikasi (*Continuous Deployment*), memastikan pengguna selalu mengakses versi terbaru.



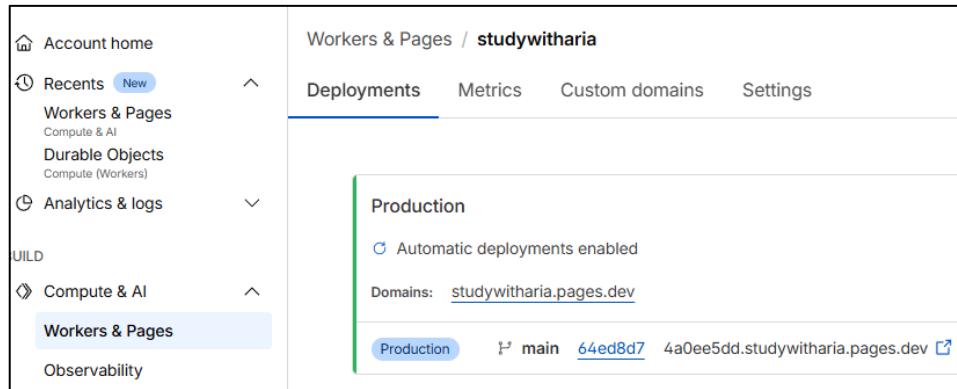
The screenshot shows a GitHub repository interface. On the left, there's a sidebar with a 'Files' section containing a tree view of files and folders. The 'js' folder is expanded, showing several JavaScript files like component-data.js, credits-data.js, guide-data.js, interaction-manager.js, loading-manager.js, main.js, model-loader.js, quiz-data.js, scene-setup.js, ui-creator.js, and vr-manager.js. Other files visible include assets, css, README.md, and favicon.ico. The main area shows the content of 'index.html'. The code editor displays the following HTML and JavaScript:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=no">
<title>WebXR Computer Lab</title>
<link rel="stylesheet" href="css/style.css" />
<link rel="icon" type="image/png" href="favicon.ico" />

<script type="importmap">
{
  "imports": {
    "three": "https://unpkg.com/three@0.162.0/build/three.module.js",
    "three/addons": "https://unpkg.com/three@0.162.0/examples/jsm/",
    "stats.js": "https://unpkg.com/three@0.162.0/examples/jsm/libs/stats.module.js"
  }
}
</script>
</head>
<body>
  <div id="orientation-overlay" class="hidden">
    <div class="message">
      <svg
        xmlns="http://www.w3.org/2000/svg"
        viewBox="0 0 24 24"
        fill="currentColor"
        width="80px"
        height="80px"
      >
        <path
          d="M18.5,6.5H15.5a1,1,0,0,1,0,0,1,1,0,0,1,0,0,1-2,0V6.92A9,9,0,0,1,6
        >
      </svg>
    </div>
  </div>
</body>

```

Gambar 4. 17 Tampilan Repositori Proyek di Github



Gambar 4. 18 Tampilan Dasbor Cloudflare Pages

2) Inisialisasi *Engine Three.js (Scene, Camera, Renderer)*

Setelah aplikasi dimuat oleh *browser* pengguna, logika inti yang terdapat dalam *main.js* dan *scene-setup.js* dieksekusi. Tahap ini bertanggung jawab untuk membangun dunia virtual sesuai arsitektur *engine* Three.js, yang mengimplementasikan bagian (*Three.js Engine*) pada Gambar 3.10. Proses ini melibatkan inisialisasi tiga komponen fundamental:

- THREE.Scene: Dibuat sebagai wadah utama untuk semua objek 3D.
- THREE.PerspectiveCamera: Dikonfigurasi sebagai mata virtual pengguna.
- THREE.WebGLRenderer: Disiapkan untuk menggambar adegan ke elemen *<canvas>* HTML.

```
// scene-setup.js: Inisialisasi komponen inti Three.js
export const scene = new THREE.Scene();
export const camera = new THREE.PerspectiveCamera(
    CAMERA_FOV,
    window.innerWidth / window.innerHeight,
    CAMERA_NEAR,
    CAMERA_FAR
);
export const renderer = new THREE.WebGLRenderer({
    antialias: true, // Menghasilkan visual yang lebih halus
});
renderer.setSize(window.innerWidth, window.innerHeight);
// Memasukkan canvas renderer ke dalam halaman HTML
document.getElementById("container").appendChild(renderer.domElement);
```

Gambar 4. 19 Inisialisasi komponen inti Three.js

Kode dari *scene-setup.js* yang menginisialisasi tiga pilar Three.js. *scene*, *camera*, dan *renderer* dibuat sebagai konstanta. *renderer* kemudian diatur ukurannya dan elemen <canvas> miliknya ditambahkan ke halaman HTML (*container*), sehingga siap menampilkan grafis.

3) Penanganan Interaksi Konsisten (*Raycaster*)

Untuk menjawab rumusan masalah mengenai interaksi yang konsisten, arsitektur aplikasi mengimplementasikan modul *interaction-manager.js*. Modul ini mengabstraksi perangkat *input* (*Mouse* dan *Controller VR*) menggunakan THREE.Raycaster. Mekanisme ini menembakkan sinar virtual dari *input* pengguna untuk mendeteksi objek 3D (tombol atau model) yang sedang ditunjuk.

```
// interaction-manager.js: Inisialisasi Raycaster
const raycaster = new THREE.Raycaster();
const pointer = new THREE.Vector2();

// Fungsi untuk deteksi di mode Desktop (Mouse)
function getIntersectedObject(x, y) {
    pointer.set(x, y); // Normalisasi koordinat mouse
    raycaster.setFromCamera(pointer, camera);
    const intersects = raycaster.intersectObjects([...], true);
    // ... (logika jika tombol terdeteksi) ...
}

// Fungsi untuk deteksi di mode VR (Controller)
function getVRIntersectedObject(controller) {
    raycaster.setFromXRController(controller);
    const intersects = raycaster.intersectObjects([...], true);
    // ... (logika jika tombol terdeteksi) ...
}
```

Gambar 4. 20 Inisialisasi *Raycaster*



Kode menunjukkan dua fungsi dari *interaction-manager.js* yang menggunakan satu *raycaster* untuk dua mode *input* berbeda. *getIntersectedObject()* menggunakan *raycaster.setFromCamera()* untuk menembakkan sinar dari posisi kursor *mouse*. Sebaliknya, *getVRIntersectedObject()* menggunakan *raycaster.setFromXRController()* untuk menembakkan sinar dari ujung *controller* VR. Keduanya mengembalikan objek yang sama, memungkinkan logika *handleInteraction()* di *main.js* tetap konsisten untuk menyesuaikan perangkat input pengguna.

4) Logika Sesi dan Integrasi WebXR

Logika inti aplikasi (*main.js*) berfungsi sebagai pengelola status. Saat pengguna menekan tombol mode VR (*Scene 4*), *main.js* memanggil fungsi *startVRSession* yang berinteraksi dengan WebXR API. Fungsi ini meminta izin kepada *browser* untuk memulai sesi *immersive*.

```
// vr-manager.js: Memulai sesi WebXR
export async function startVRSession(onSessionEndCallback,
...) {
  try {
    // Meminta sesi 'immersive-vr' dari browser
    const session = await
navigator.xr.requestSession("immersive-vr", {
      optionalFeatures: ["local-floor"],
    });

    // Memberitahu renderer Three.js untuk menggunakan
    // sesi VR ini
    renderer.xr.setSession(session);

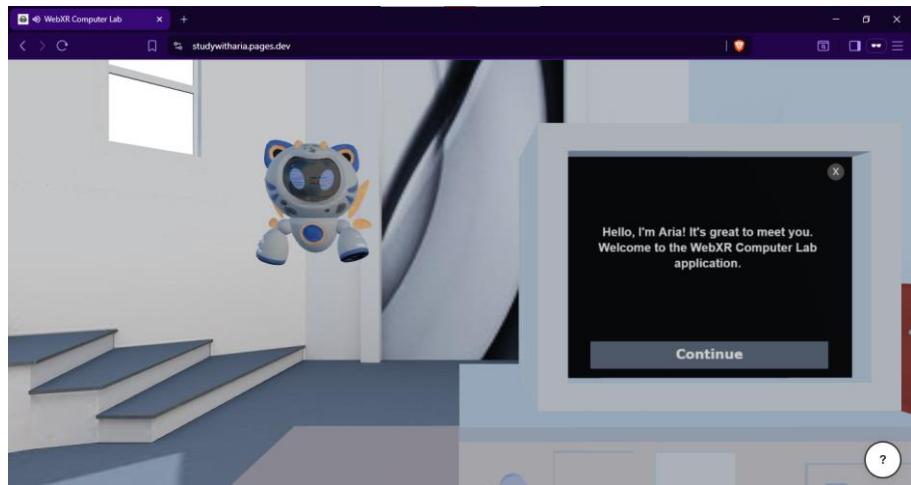
    session.addEventListener("end", onSessionEndCallback);
  } catch (e) {
    // ... (Penanganan error jika VR tidak tersedia) ...
  }
}
```

Gambar 4. 21 Konfigurasi sesi *Virtual Reality* (VR)

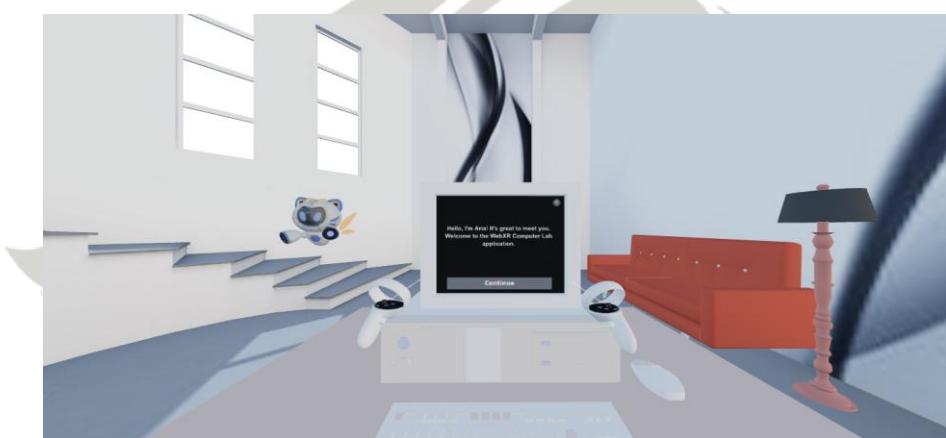
Kode menunjukkan implementasi WebXR API dari *vr-manager.js*. Fungsi *startVRSession()* menggunakan perintah *navigator.xr.requestSession ("immersive-vr")* untuk memulai mode VR. Jika berhasil, sesi VR yang didapat akan diserahkan ke *renderer* Three.js (*renderer.xr.setSession*), yang secara otomatis akan beralih ke mode *rendering* VR.

5) Proses Render dan Keluaran (*Render Loop* dan *Output*)

Seluruh proses di atas disatukan dalam *render loop* menggunakan *rendering engine* WebGL API. Ini adalah fungsi *render()* di *main.js* yang dipanggil secara terus-menerus. Fungsi ini memberi instruksi kepada GPU untuk memproses grafis dan menampilkannya di perangkat *output*



Gambar 4. 22 Tampilan Hasil *Render Mode Desktop*



Gambar 4. 23 Tampilan Hasil *Render Mode VR*

```
// main.js: Fungsi render loop utama
function animate() {
    renderer.setAnimationLoop(render);
}

function render() {
    // ... (Logika update animasi, interaksi VR, dll.) ...

    if (isVRMode()) {
        handleVRHover(); // Update raycaster VR
        handleVRDrag(deltaTime); // Update rotasi model via VR
    } else {
        controls.update(); // Update orbit controls mouse
    }

    // Perintah akhir untuk menggambar ke layar
    renderer.render(scene, camera);
}
```

Gambar 4. 24 Mekanisme *Render* dan *Update Frame*

Kode menunjukkan fungsi `render()` dari `main.js`. Fungsi `animate()` mengatur agar `render()` dipanggil setiap *frame*. Di dalam `render()`, logika *update* yang spesifik untuk mode VR, seperti `handleVRHover()` atau mode *desktop* seperti `controls.update` dijalankan. Baris terakhir, `renderer.render(scene, camera)`, adalah instruksi final yang mengirimkan seluruh adegan ke GPU untuk digambar ke perangkat *output* (monitor atau *headset VR*).

4.4 Hasil Pengujian

Pengujian dilakukan untuk memvalidasi aplikasi sesuai rancangan Bab III, mencakup verifikasi fungsional (*black box*), pengukuran performa, dan evaluasi usabilitas dengan *System Usability Scale* (SUS).

4.4.1 Hasil Pengujian Fungsional

Pengujian fungsional dilakukan untuk memverifikasi bahwa setiap fitur aplikasi berjalan sesuai dengan kebutuhan fungsional yang didefinisikan pada Tabel 3.1. Pengujian ini mengikuti skenario yang telah disusun pada Tabel 3. 13 dan diulang sebanyak 10x, di mana peneliti menjalankan serangkaian aksi dan membandingkan hasil yang ditampilkan sistem dengan hasil yang diharapkan. Hasil dari pengujian fungsional dirangkum pada Tabel 4. 7.

Tabel 4. 7 Hasil Pengujian Fungsional *Black Box*

No	Fitur yang Diuji	Skenario Aksi	Hasil yang Diharapkan	Status
1	Alur Awal Aplikasi	Memulai aplikasi, memasukkan nama, dan memilih mode (<i>Browser</i> atau <i>VR</i>).	Aplikasi menampilkan layar selamat datang, menyimpan nama pengguna, dan berhasil masuk ke menu utama sesuai mode yang dipilih.	Berhasil
2	Navigasi Menu	Dari menu utama, memilih "Mulai Belajar" dan "Laporan Belajar".	Sistem menampilkan halaman pemilihan materi saat "Mulai Belajar" dipilih dan menampilkan halaman laporan saat "Laporan Belajar" dipilih.	Berhasil
3	Tampilan & Interaksi Model	Memilih salah satu materi, melakukan kontrol kamera dan memutar model 3D.	Model 3D komponen yang dipilih ditampilkan dengan benar dan dapat diputar oleh pengguna serta sudut pandang kamera dapat diubah oleh pengguna.	Berhasil

Tabel 4. 8 Hasil Pengujian Fungsional *Black Box* (Lanjutan)

No	Fitur yang Diajukan	Skenario Aksi	Hasil yang Diharapkan	Status
4	Informasi & Audio	Menampilkan panel informasi dan menekan tombol audio.	Panel informasi menampilkan teks penjelasan yang sesuai. Audio penjelasan dapat diputar saat tombol ditekan.	Berhasil
5	Mini Kuis	Menyelesaikan satu materi dan menjawab mini kuis.	Setelah materi selesai, mini kuis muncul. Setelah dijawab, sistem memberikan umpan balik (benar/salah) dan membuka materi selanjutnya jika jawaban benar.	Berhasil
6	Tes Akhir	Menyelesaikan semua materi dan mengerjakan tes akhir.	Tombol "Tes Akhir" menjadi aktif. Pengguna dapat mengerjakan soal hingga selesai.	Berhasil
7	Laporan Belajar	Menyelesaikan tes akhir dan mengakses menu "Laporan Belajar".	Laporan menampilkan skor akhir tes dengan benar.	Berhasil
8	Mode VR	Menekan tombol "Mode VR".	Aplikasi berhasil beralih ke mode VR tanpa error, dan interaksi dapat dilakukan menggunakan <i>controller</i> VR.	Berhasil
9	Kompatibilitas <i>Browser</i>	Menjalankan aplikasi di browser Chrome, Firefox, dan Edge.	Aplikasi berjalan dengan normal dan semua fungsionalitas dapat digunakan di ketiga browser tersebut.	Berhasil

4.4.2 Hasil Pengujian Performa

Pengujian performa dilakukan untuk memvalidasi secara kuantitatif apakah aplikasi telah memenuhi kebutuhan Non Fungsional, khususnya terkait waktu muat aset dan kelancaran *frame rate* (FPS). Pengujian ini dilaksanakan sesuai dengan skenario dan spesifikasi perangkat yang telah ditetapkan pada Tabel 3. 15.

Pertama, pengukuran waktu muat bertujuan untuk memverifikasi bahwa aplikasi dapat diakses dengan cepat. Skenario pengujian ini menyimulasikan kondisi jaringan pada kecepatan maksimal 10 Mbps menggunakan fitur pada Chrome DevTools. Hasil pengukuran waktu yang dibutuhkan untuk mengunduh setiap aset 3D .glb disajikan pada Tabel 4.9.

Tabel 4. 9 Hasil Uji Performa Pengukuran Waktu Muat Model 3D

No	Komponen	Waktu Muat (detik)
1	Monitor	1,41
2	Keyboard	1,87
3	Mouse	0,154
4	Printer	0,120
5	Storage	0,321
6	Motherboard	2,43
7	CPU	2,05
8	Kartu Jaringan	0,514
9	RAM	1,31
10	GPU	0,983
11	Flashdisk	0,890
12	Avatar	0,771
13	Ruangan	0,209
Rata-Rata		1,05

Berdasarkan Tabel 4.9, waktu muat rata-rata seluruh aset 3D adalah 1,05 detik. Angka ini telah memenuhi kriteria keberhasilan pengujian waktu muat, yang menetapkan ambang batas waktu muat di bawah 5 detik. Meskipun model *motherboard* mencatatkan waktu muat terlama, yaitu 2,43 detik, nilai tersebut masih berada di dalam batas toleransi.

Kemudian, Pengukuran FPS bertujuan untuk memastikan aplikasi berjalan dengan lancar dan memberikan pengalaman visual yang nyaman. Pengukuran dilakukan menggunakan pustaka *stats.module.js* pada skenario interaksi normal dan skenario uji beban dengan memuat model 3D avatar di halaman tutorial. Hasil observasi dirangkum pada Tabel 4.10.

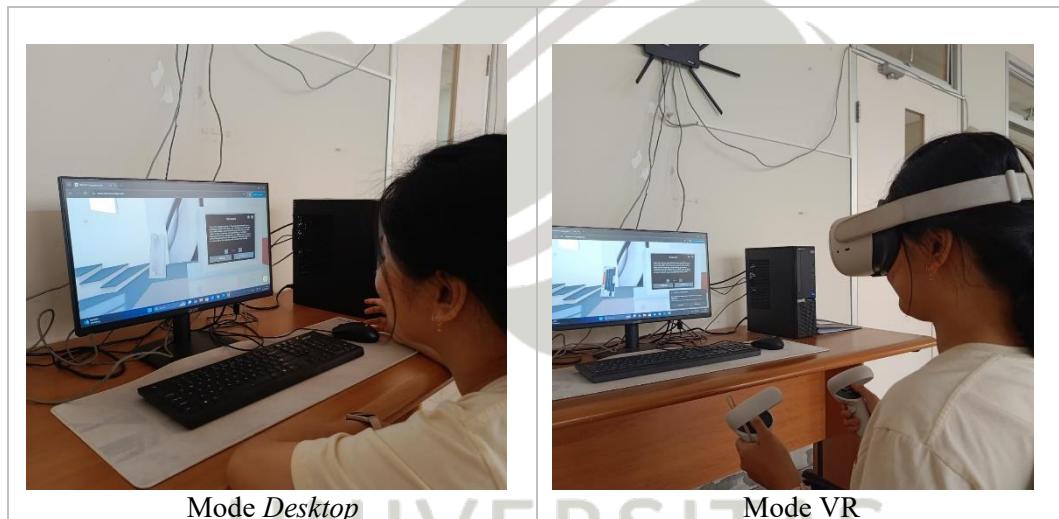
Tabel 4. 10 Hasil Uji Performa Pengukuran FPS

No	Mode Pengujian	Skenario Uji	Hasil Observasi (FPS)
1	Mode Desktop	Interaksi normal dengan navigasi UI, rotasi kamera, serta pemutaran model 3D selama 60 detik.	Stabil di 58-60 FPS.
2	Mode Desktop	Uji beban dilakukan saat memuat model 3D <i>avatar</i> di halaman tutorial, yang merupakan model dengan tingkat kompleksitas tertinggi.	Penurunan sesaat ke 55 FPS, lalu stabil di 58-60 FPS.

Tabel 4. 11 Hasil Uji Performa Pengukuran FPS (Lanjutan)

No	Mode Pengujian	Skenario Uji	Hasil Observasi (FPS)
3	Mode VR	Interaksi normal dengan navigasi UI dan rotasi model menggunakan <i>controller</i> VR selama 60 detik.	FPS kurang stabil di 72-90. Penurunan terjadi ketika ada interaksi memuat komponen perangkat keras.
4	Mode VR	Uji beban saat memuat model 3D <i>avatar</i> pada halaman tutorial dalam mode VR.	Penurunan FPS pada saat pemuatan awal hingga di 55 FPS, lalu berjalan di 73-90 FPS pada halaman ini.

4.4.3 Hasil Pengujian Kegunaan (*Usabilitas*)



Gambar 4. 25 Implementasi Penggunaan Aplikasi pada Mode *Desktop* dan VR

Pengujian kegunaan (usabilitas) dilakukan untuk mengukur tingkat kemudahan penggunaan dan kepuasan pengguna terhadap aplikasi secara subjektif. Pengujian ini menggunakan metode *System Usability Scale* (SUS) dan melibatkan 30 partisipan yang sesuai dengan target pengguna. Partisipan diminta untuk mencoba aplikasi dan menyelesaikan keseluruhan materi, kemudian mengisi kuesioner yang terdiri dari 10 pernyataan. Hasil perolehan kuesioner dari setiap responden disajikan pada Lampiran D dan hasil pengujian usabilitas (SUS) dapat dilihat pada Tabel 4. 12.

Tabel 4. 12 Hasil Pengujian Usabilitas (SUS)

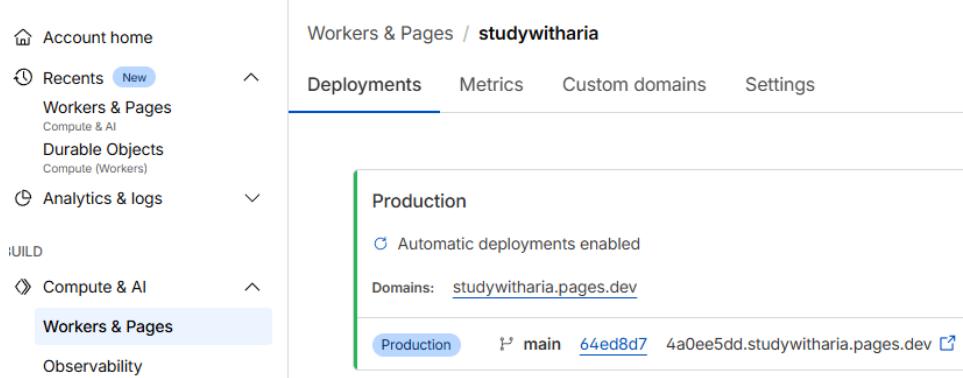
R	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	JML	Nilai (JML x 2,5)
1	3	4	4	2	4	3	4	4	4	3	35	87.5
2	3	3	4	4	4	4	4	4	4	4	38	95
3	4	3	4	4	3	3	4	3	3	3	34	85
4	3	3	3	2	3	3	4	3	3	3	30	75
5	2	3	3	3	3	3	2	4	3	4	30	75
6	3	4	4	3	4	3	3	4	3	3	34	85
7	4	4	4	4	3	3	4	4	3	3	36	90
8	3	3	4	3	4	4	3	4	4	3	35	87.5
9	2	2	3	1	3	3	3	3	2	2	24	60
10	3	4	4	3	4	3	4	3	4	0	32	80
11	3	0	4	2	3	3	4	4	3	3	29	72.5
12	3	1	4	2	4	3	4	4	4	1	30	75
13	2	4	4	4	4	0	4	4	4	4	34	85
14	4	4	4	4	4	4	4	4	4	0	36	90
15	3	4	4	3	3	3	4	4	3	3	34	85
16	2	4	4	2	2	3	2	2	4	4	29	72.5
17	3	3	4	3	3	4	4	4	3	3	34	85
18	4	2	2	1	3	2	2	2	4	2	24	60
19	3	3	3	2	3	2	4	3	3	2	28	70
20	3	4	4	2	3	3	3	3	3	2	30	75
21	3	3	3	2	3	4	4	3	3	2	30	75
22	3	4	3	4	4	4	4	4	3	4	37	92.5
23	3	3	3	2	3	3	3	3	3	3	29	72.5
24	4	3	4	1	4	4	4	4	3	3	34	85
25	4	1	3	2	4	3	4	3	3	3	30	75
26	3	4	4	3	4	3	3	4	3	3	34	85
27	0	3	4	2	4	4	3	3	2	2	27	67.5
28	4	3	4	1	4	3	4	4	3	3	33	82.5
29	2	3	2	1	3	3	2	3	2	2	23	57.5
30	3	4	4	2	4	3	4	4	4	1	33	82.5
Skor Rata-rata (Hasil Akhir)											31.53	78.83

Berdasarkan Tabel 4.12, skor rata-rata SUS adalah 78,83, berada di atas ambang 68, sehingga usabilitas aplikasi dinilai *acceptable*. Skor ini juga mendekati 80, menunjukkan tingkat penerimaan pengguna yang kuat terhadap kemudahan penggunaan aplikasi.

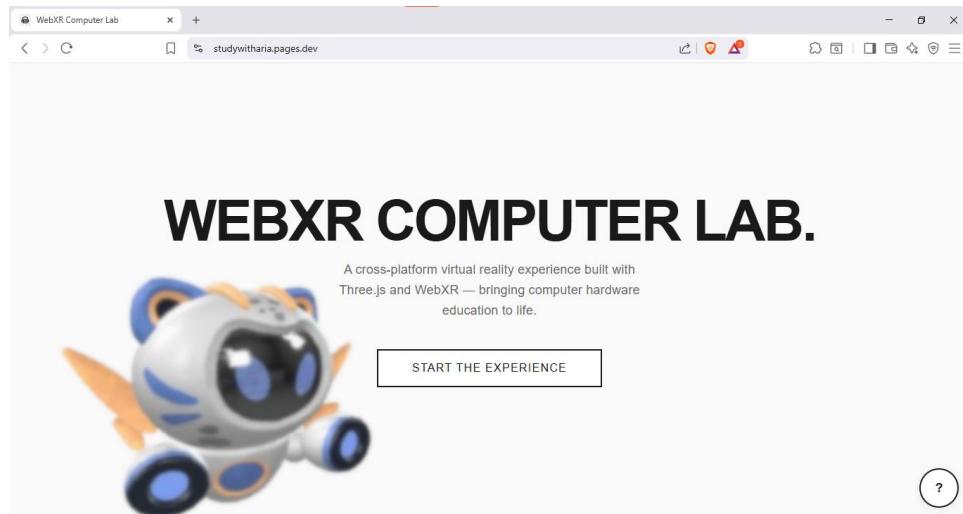
4.5 Hasil Distribusi

Tahap distribusi merupakan implementasi dari langkah terakhir dalam metodologi MDLC, yaitu proses publikasi aplikasi agar dapat diakses oleh pengguna sasaran. Sesuai dengan rancangan arsitektur, aplikasi *Virtual Reality Pengenalan Perangkat Keras Komputer* telah didistribusikan secara daring dengan memanfaatkan dua platform utama. Platform tersebut adalah GitHub yang digunakan sebagai repositori untuk pengelolaan kode sumber dan aset proyek, serta Cloudflare Pages yang berfungsi sebagai layanan *hosting* untuk penyebaran aplikasi.

Proses distribusi ini menerapkan alur kerja *Continuous Deployment* (CD) yang terintegrasi antara Cloudflare Pages dan *branch main* pada repositori GitHub. Setiap pembaruan kode yang digabungkan ke branch utama secara otomatis akan memicu proses kompilasi (*build*) dan penerapan (*deployment*) yang baru. Melalui mekanisme tersebut, versi aplikasi yang tersedia secara publik selalu merupakan versi terkini tanpa memerlukan intervensi manual, sehingga konsistensi dan efisiensi dalam pemeliharaan aplikasi dapat terjaga.



Gambar 4. 26 Tampilan Dasbor Distribusi Cloudflare Pages



Gambar 4. 27 Hasil Distribusi Aplikasi

Hasil akhir dari tahap distribusi ini adalah aplikasi yang aktif, stabil, dan dapat diakses oleh pengguna secara global melalui *browser* yang mendukung WebXR. Aplikasi ini secara otomatis dilayani melalui koneksi aman (HTTPS) untuk menjamin keamanan data pengguna. Aplikasi dapat diakses pada tautan berikut: <https://studywitharia.pages.dev/>.

UNIVERSITAS
MA CHUNG

Bab V

Simpulan dan Saran

Bab ini berisi simpulan akhir yang ditarik dari hasil analisis dan pembahasan yang telah disajikan pada bab-bab sebelumnya. Simpulan ini dirumuskan untuk menjawab secara langsung perumusan masalah yang telah ditetapkan. Selain itu, bab ini juga menyertakan saran-saran untuk pengembangan aplikasi di masa depan berdasarkan temuan dan keterbatasan selama penelitian.

5.1 Simpulan

Berdasarkan hasil implementasi dan pengujian aplikasi *Virtual Reality* (VR) pengenalan perangkat keras komputer menggunakan Three.js dan WebXR, dapat ditarik beberapa simpulan sebagai berikut:

- 1) Implementasi Three.js dan WebXR terbukti stabil. Aplikasi berjalan lancar pada mode *desktop* dengan 60 FPS dan pada mode VR dengan kisaran 72–90 FPS. Penurunan performa hanya terjadi sesaat saat memuat materi berat menjadi 55 FPS, namun tidak mengganggu pengalaman keseluruhan.
- 2) Alur kerja optimasi model 3D berhasil mereduksi ukuran asset sebesar 84,97%, penggunaan VRAM sebesar 80,40%, dan jumlah *draw call* sebesar 71,21%. Alur kerja ini mengombinasikan reduksi *vertex*, kompresi tekstur KTX2, dan kompresi geometri Draco
- 3) Mekanisme interaksi yang konsisten antara mode *desktop* dan VR berhasil dicapai melalui abstraksi input menggunakan *THREE.Raycaster*. Metode ini menerjemahkan input dari kurSOR *mouse* dan *controller* VR ke dalam satu logika penanganan interaksi yang seragam, sehingga fungsionalitas navigasi dan seleksi objek tetap konsisten di kedua mode.

Secara keseluruhan, penelitian ini membuktikan bahwa tumpukan teknologi Three.js dan WebXR merupakan alternatif yang layak untuk pengembangan aplikasi VR edukasi berbasis web. Pendekatan ini menawarkan keunggulan dalam hal aksesibilitas, bobot aplikasi yang ringan, dan kebebasan dari model lisensi

berbayar jika dibandingkan dengan platform dominan seperti Unity. Aplikasi yang dihasilkan fungsional dan memiliki tingkat usabilitas yang dapat diterima, dibuktikan dengan skor SUS sebesar 78,83.

5.2 Saran

Meskipun penelitian ini telah berhasil mencapai tujuannya, terdapat beberapa keterbatasan dan peluang pengembangan yang dapat dijadikan acuan untuk penelitian selanjutnya:

- 1) Meskipun *frame rate* (FPS) saat ini sudah stabil, disarankan untuk melakukan optimalisasi lebih lanjut agar pengalaman pengguna di *headset VR* menjadi lebih lancar dan nyaman, serta mengurangi potensi *motion sickness*.
- 2) Salah satu batasan penelitian ini adalah interaksi yang terbatas pada melihat, memutar, dan evaluasi kuis. Pengembangan selanjutnya dapat meningkatkan imersi dengan menambahkan interaktivitas yang lebih kompleks, seperti simulasi perakitan komputer.
- 3) Konten aplikasi dapat diperkaya dengan menambahkan lebih banyak materi perangkat keras komputer. Selain itu, variasi soal evaluasi, selain kuis benar/salah dan pilihan ganda, dapat ditambahkan untuk mengukur pemahaman pengguna dengan cara yang berbeda.
- 4) Mengingat WebXR juga mendukung AR, penelitian selanjutnya dapat mengembangkan mode AR. Fitur ini akan memungkinkan pengguna memproyeksikan model 3D perangkat keras seolah-olah berada di atas meja belajar mereka secara nyata menggunakan kamera *smartphone*.

DAFTAR PUSTAKA

- Abuhamra, A.E., 2023, Importance of Computer Hardware, IJAEM: International Journal of Advances in Engineering and Management, 5:428-433.
- Andysa, S., 2022, Mengenal System Usability Scale, School of Information Systems Binus University, diakses pada 4 November 2025, <https://sis.binus.ac.id/2022/02/07/mengenal-system-usability-scale/>.
- Anggoro, N.A. and Astuti, I.A., 2024, Development of Virtual Reality Application for Arachnophobia Using Multimedia Development Life Cycle Method, Journal of Information Systems and Informatics, 6:1535-1558.
- Astianingrum, K., Sunardi and Herman., 2025, Proses Rendering Model 3D Jantung Manusia pada Aplikasi Web Mobile Menggunakan WebGL, JATI (Jurnal Mahasiswa Teknik Informatika),9:2527-2533.
- Azmi, M.N., Mansur, H. and Utama, A.H., 2024, Potensi Pemanfaatan Virtual Reality Sebagai Media Pembelajaran di Era Digital, JDPP: Jurnal Dimensi Pendidikan dan Pembelajaran, 12:211-226.
- Buchori, A., Cahya, P. Z., Wardani, T. I., Khoiri, N. and Osman, S. 2025, Design and Development of an Interactive Scouting Educational Game Using the MDLC Approach, ASSET: Advance Sustainable Science, Engineering and Technology, 7(2), 02502023-01~02502023-08
- Budiarti, R.P., Sulistyani, E., Anggraini, F., Marvyna, H. C. and Manilaturrohmah, 2022, Pengenalan Teknologi Virtual Reality dan Augmented Reality Untuk Meningkatkan Pembelajaran Digital di Pondok Pesantren Al Muin Syarif Hidayatullah, Communautaire: Journal of Community Service, 1:90-97.
- CGTyphoon, 2025, Vertices, edges and faces of 3D object, CGTyphoon, diakses pada 5 Desember 2025, <https://cgtphoon.com/fundamentals/vertices-edges-and-faces-of-3d-object/>.
- Cloudflare Docs, 2025, Cloudflare Pages - Create full-stack applications that are instantly deployed to the Cloudflare global network, Cloudflare Docs, diakses pada 5 November 2025, <https://developers.cloudflare.com/pages/>.
- De Luna, A.A., 2022, *Introduction to Virtual Reality*, Arcler Press, Burlington, ON, Canada.
- Deriota, 2023, Mengenal GitHub dan Cara Penggunaannya, Deriota, diakses pada 10 September 2025, <https://deriota.com/news/read/1119/mengenal-github-dan-cara-penggunaanya.html>.
- Erkamim, M., 2023, *Pengantar Sistem Informasi*, PT Prime Identity House, Ponorogo.

- Fachri, M. and Darmawan, R., 2022, Visualisasi Model 3D Dinamis Berbasis Web Menggunakan WebGL, Syntax Literate: Jurnal Ilmiah Indonesia, 7(10):15301-15310.
- Fanini, B., Ferdani, D., Demetrescu, E., Berto, S. and Annibale, E., 2021, ATON: An Open-Source Framework for Creating Immersive, Collaborative and Liquid Web-Apps for Cultural Heritage, Applied Sciences, 11(22):1-38.
- Fitriyanto, M.N., Rozikin, A.Z., Leuwincun, L., Sudianto, N. and Astuti, A.D., 2024, Pemanfaatan Teknologi Virtual Reality dalam Pengenalan Sel Tubuh Manusia, HUMA BETANG: Jurnal Pengabdian Masyarakat, 1:17-25.
- Hardiana, Akramunnisa and Jumarniati., 2025, Rancang Bangun Aplikasi Multimedia Pengenalan Perangkat Keras Komputer Berbasis Augmented Reality, Artificial: Jurnal Informatika dan Sistem Informasi, 3:186-200.
- Hussain, F., 2020, Unity Game Development Engine: A Technical Survey, USJICT: University of Sindh Journal of Information and Communication Technology, 4(2):73-81.
- Irawan, P.L.T. and Rahmadianto, S.A., 2025. Ma Chung Virtual Campus Tour Berbasis Virtual Reality. JEPIN: Jurnal Edukasi dan Penelitian Informatika, 11(2): 337-342.
- Kurniasari, A.A., Puspitasari, T.D. and Mutiara, A.S., 2023, Penerapan Metode Multimedia Development Life Cycle (MDLC) pada A Magical Augmented Reality Book Berbasis Android, ANTIVIRUS: Jurnal Ilmiah Teknik Informatika, 17:19-31.
- Kurniawan, D., 2022, *Belajar Pemrograman Web Dasar HTML, CSS, & Javascript untuk Pemula*, Yayasan Prima Agus Teknik, Semarang.
- Landa, V. R. ., Djusmin, V. B. and Ekawati, S., 2025, Pengembangan Media Pembelajaran Sistem Pencernaan Manusia Berbasis Augmented Reality untuk Siswa SMPN 2 Palopo, Jurnal Literasi Digital, 5: 96–109.
- Lee, T., 2024, Unity Engine Dissection: Improvement Points in Comparison between Unity Engine and Open-Source Engines, Asia-pacific Journal of Convergent Research Interchange, 10(11):449-458.
- Matahari, T., 2022, WebXR Asset Management in Developing Virtual Reality Learning Media, IJoCED: Indonesian Journal of Computing, Engineering, and Design, 4(1):38-46.
- Meta, 2024, Loading custom 3D models with glTF - Meta Spatial SDK Documentation, Meta, diakses pada 5 November 2025, <https://developers.meta.com/horizon/documentation/spatial-sdk/spatial-sdk-gltf/>.

- McCurdy, D., 2025, glTF 2.0 SDK for JavaScript and TypeScript, on Web and Node.js, glTF Transform, diakses pada 15 September 2025, <https://gltf-transform.dev/>.
- Moioli, G., 2022, *Introduction to Blender 3.0: Learn Organic and Architectural Modeling, Lighting, Materials, Painting, Rendering, and Compositing with Blender*, Apress, Milano.
- Mouattalib, H., Tabaa, M. and Youssefi, M., 2023, Revolutionizing Engineering Education: Creating a Web-Based Teaching Platform for Immersive Learning Experiences, *Journal of Smart Cities and Society*, 2(2):151-162.
- Mozilla Mixed Reality, 2024, WebXR emulator extension, Github, diakses pada 10 September 2025, <https://github.com/MozillaReality/WebXR-emulator-extension>.
- Nur, H., Jusrianto, Muis, I. and Anas, A., 2024, Pengembangan Media Pembelajaran Berbasis Virtual Reality untuk Membentuk Karakter Peduli Lingkungan Siswa Tingkat Sekolah Dasar, *Didaktika: Jurnal Kependidikan*, 13(4):5337-5346.
- Nuswantoro, F., Kurniawan, E. and Totti, D.F., 2024, Perbandingan Kualitas dan Efisiensi Render antara Eevee dan Cycles Blender dalam Film Animasi Ireng, *Jurnal Imaji: Film, Fotografi, Televisi, dan Media Baru*, 15(3):204-215.
- Prasatya, 2024, Apa Itu Black Box Testing? Pengertian, Contoh, dan Manfaatnya untuk Teknik Pengujian Software, Codepolitan, diakses pada 4 November 2025, <https://www.codepolitan.com/blog/apa-itu-black-box-testing-pengertian-contoh-dan-manfaatnya-untuk-teknik-pengujian-software/>.
- Purwoko, N.E. and Zen, B.P., 2023, Aplikasi Pembelajaran Bangun Ruang Menggunakan Augmented Reality Marker Based Tracking, *Jurnal Ilmiah MEDIA SISFO*, 99:302-312.
- Putra, D.A., Adisusilo, A.K. and Prasetya, N.I., 2022, Optimasi Aset dan Karakter Permainan 3D Berbasis Tematik Sekolah Dasar, *Journal of Information System, Graphics, Hospitality and Technology*, 4(1):1-6.
- Rantamaa, H. R., Kangas, J., Kumar, S. K., Mehtonen, H., Jarnstedt, J., and Raisamo, R., 2023, Comparison of a VR Stylus with a Controller, Hand Tracking, and a Mouse for Object Manipulation and Medical Marking Tasks in *Virtual Reality*, *Applied Sciences*, 13(4):1-17.
- Repiliya and Mair, Z.R., 2021, Computer Hardware Recognition Application with Augmented Reality Technology, *Cyberspace: Jurnal Pendidikan Teknologi Informasi*, 5:59-68.
- Sani, F., M. 2020, Pengertian dan Teknik Model 3D yang Harus Kamu Ketahui, GameLab, diakses pada 5 Desember 2025,

- <https://www.gamelab.id/news/231-pengertian-dan-teknik-model-3d-yang-harus-kamu-ketahui#:~:text=Vertex%20merupakan%20titik%20yang%20terletak,utuh%20yang%20disebut%20dengan%20Mesh..>
- Srivastava, A., 2025, GPT-SoVITS: A Practical Guide to Voice Synthesis using Python, Medium, diakses pada 5 November 2025, <https://medium.com/@akshit0405/gpt-sovits-a-practical-guide-to-voice-synthesis-using-python-cce78818c933>.
- Straccia, A., 2024, *Interactive Web Development With Three.js and A-Frame*, Orange Education Pvt Ltd, Delhi.
- Threkit, 2025, Performance Guidelines, Threkit, diakses pada 12 Desember 2025, https://community.threkit.com/learn/workflows/best-practices/performance-considerations?utm_source=chatgpt.com&q=webgl.
- Wibowo, A., 2022, *Kekuatan Augmented dan Virtual Reality dalam Bisnis – Jilid 1*, Yayasan Prima Agus Teknik, Semarang.
- Xu, K., 2024, OpenVINO Enable Digital Human-TTS (GPT-SoVITs), OpenVINO, diakses pada 5 November 2025, <https://blog.openvino.ai/blog-posts/openvino-enable-digital-human-tts-gpt-sovits>.
- Zhang, W., Li, J., Ji, L., Cheng, X., Sun, D., Jiang, Y., Chen, F., Zhou, Y., Choi, C., Cheng, H. and Cai, S., 2025. FNIRS experimental study on the impact of AI-synthesized familiar voices on brain neural responses. *Scientific Reports*, 15:1-16.
- Zhuk, Y. and Tetuev, E., 2020, 'Prospects of WebGL Technology Using in Modern Educational Process', CEUR Workshop Proceedings, *Saint-Petersburg State Forestry University*, 2630:112-122.

UNIVERSITAS
MA CHUNG

Lampiran A
Naskah Narasi Audio

- 1) Introduction: The history of computers began with simple counting tools such as the abacus thousands of years ago, then developed into mechanical calculating machines in the 17th century. The first electronic computers, such as ENIAC in 1946, were room-sized and used thousands of vacuum tubes. The development of transistors in the 1950s allowed computers to become smaller and more efficient, followed by integrated circuits in the 1960s that further reduced computer size. The era of personal computers began in the 1970s with the emergence of the Apple II and IBM PC, making computers usable in homes and small offices. The 1980s to 1990s became a period of rapid development with the arrival of Windows operating systems, the mouse, and user-friendly graphical interfaces. The Internet became popular in the 1990s, changing the way humans communicate and access information. The 21st century brought the revolution of mobile computers with increasingly thin and light laptops, smartphones as pocket computers, and tablets combining portability with touchscreens. Cloud computing technology enables data access from anywhere, while artificial intelligence and machine learning open a new era of intelligent computing that can learn and adapt. Today, quantum computers are being developed to solve extremely complex computational problems.
- 2) Monitor: A monitor is the computer's display screen that functions as a visual window between the user and the computer system. This device displays all information, images, videos, and program interfaces running on the computer. Without a monitor, we cannot see what the computer is working on. There are several types of monitors based on their technology, such as LED monitors which are most commonly used today because they are energy-efficient and provide good image quality, OLED monitors with high contrast and more vivid colors, and gaming monitors with high refresh rates for smoother gaming experiences. Monitors are also available in various sizes ranging from 15 inches to 32 inches or larger.

- 3) Keyboard: A keyboard is the main input device that allows users to enter text, numbers, and commands into the computer. Each key on the keyboard represents a specific character or function that is sent to the computer when pressed. Modern keyboards generally have 104 standard keys including letters, numbers, symbols, and special function keys. There are various types of keyboards available, from membrane keyboards that use a flexible layer and are more affordable, mechanical keyboards that provide better tactile response and durability, to wireless keyboards that use Bluetooth or a USB receiver. Some special keyboards are also equipped with backlighting or macro keys for gaming and productivity needs.
- 4) Mouse: A mouse is a pointing device that allows the user to control the cursor on the computer screen and interact with visual elements such as icons, menus, and buttons. The mouse works by detecting movement and converting it into cursor movement on the screen. This device has left and right buttons for clicking, as well as a scroll wheel for page navigation. Based on detection technology, there are optical mice that use LED sensors and can be used on almost any surface, laser mice that are more accurate and sensitive, and gaming mice with high DPI for precise movement. Mice are also available in wired and wireless versions, with ergonomics designed for long-term comfort.
- 5) Motherboard: The motherboard is the main component that connects all parts of the computer into one complete system. As the name suggests, the motherboard functions as the “mother” that provides communication pathways and electrical power for all other components such as the processor, RAM, graphics card, and storage devices. Without a motherboard, these components cannot communicate with each other. Motherboards come in various sizes or form factors, from large ATX boards suitable for desktop computers with many expansion slots, Micro-ATX which is more compact but still feature-rich, to Mini-ITX which is very small for space-saving PC builds. Each motherboard has a specific processor socket, RAM slots, input-output ports, and a chipset that determines compatibility with other components.

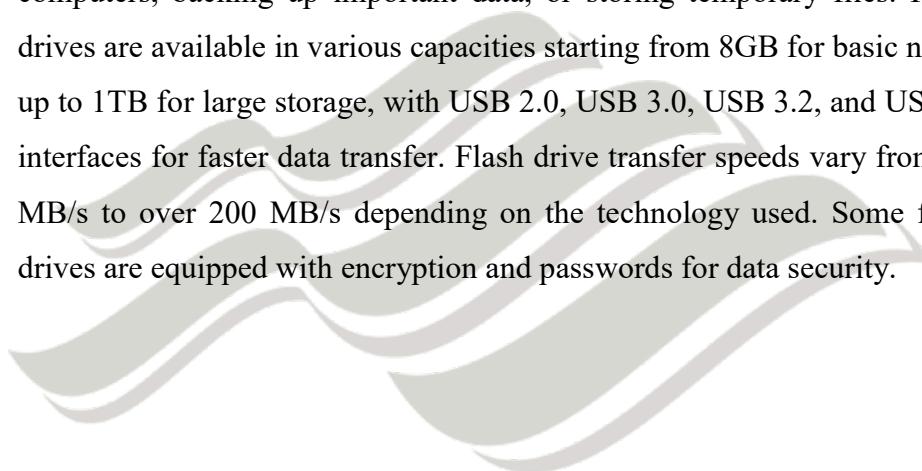
- 6) Processor (CPU): The CPU or processor is the brain of the computer that is responsible for executing all instructions and calculations required by the system and applications. Every time you open a program, type, or perform any activity on the computer, the CPU processes these commands at very high speed. CPU performance greatly determines how fast the computer can complete tasks. Modern CPU come in various core configurations, ranging from dual-core for basic needs, quad-core for smooth multitasking, to 8, 16, or even 32 cores for professional tasks such as video editing and rendering. The two main CPU manufacturers are Intel with its Core i3, i5, i7, and i9 series, and AMD with its Ryzen 3, 5, 7, and 9 series. The higher the number, the more powerful the performance offered.
- 7) Memory (RAM): Memory is a storage area used by the computer to run programs and the operating system. RAM or Random Access Memory serves as the computer's workspace, storing data that is actively in use so it can be quickly accessed by the processor. The larger the RAM capacity, the more programs can run simultaneously without slowing the computer down. ROM or Read-Only Memory is a permanent type of memory that stores basic instructions for starting the computer, such as the BIOS or UEFI. RAM is available in various capacities starting from 4GB for basic needs, 8GB for normal use, up to 16GB or 32GB for professional requirements. RAM also comes in different speeds, such as DDR4 and the newer DDR5 with better performance.
- 8) Graphics Card (GPU): The GPU or graphics card is the component responsible for processing and displaying images, videos, animations, and visual effects on the computer screen. Unlike the CPU, which is designed for general processing, the GPU has thousands of small cores that work in parallel to handle complex graphic calculations efficiently. The GPU is essential for gaming, graphic design, video editing, and applications that require high visualization. There are two types of GPUs: integrated graphics, which are built into the processor and suitable for basic needs, and dedicated graphics cards, which are separate cards with much higher performance. Dedicated GPUs are produced by NVIDIA with its GeForce

RTX and GTX series, and AMD with its Radeon RX series. Modern GPUs also support ray tracing technology for realistic lighting and DLSS to increase frame rates in gaming.

- 9) Network Card (NIC): A network card or Network Interface Card is a component that allows the computer to connect to the internet or a local network. This device acts as a communication bridge between the computer and a router, modem, or other network devices. The network card converts digital data into signals that can be transmitted via cable or radio waves for wireless connections. Network cards are available in two main types: Ethernet cards for wired connections with stable speed and low latency, and WiFi cards for wireless connections that provide mobility flexibility. Network card speeds range from 100 Mbps for basic needs, 1 Gbps for home use, up to 10 Gbps for servers and enterprise needs. Most modern motherboards already have an integrated network card.
- 10) Storage: Storage is the device that permanently stores all data, files, programs, and the operating system on the computer. Unlike RAM, which is temporary, data in storage remains saved even when the computer is turned off. Storage plays a vital role in keeping photos, videos, documents, games, applications, and all your digital files. There are two main types of storage: HDD or Hard Disk Drive, which uses rotating magnetic disks with large capacity and affordable prices, and SSD or Solid State Drive, which uses flash memory chips with much faster read-write speeds but is more expensive. Storage is available in various capacities from 128GB to several terabytes, with SATA interfaces for traditional HDDs and NVMe for modern SSDs that are super fast.
- 11) Printer: A printer is an output device that functions to print digital documents into physical form on paper or other media. The printer converts text, images, or documents from the computer into printed results that can be held and read directly. This device is very important for office, school, and household needs in producing physical documents. Based on printing technology, there are inkjet printers that use liquid ink and are suitable for printing high-quality photos but have higher operating costs, laser printers

that use powdered toner with high print speeds and are ideal for large-volume text documents, and dot matrix printers that are still used for official documents because they can print with pressure. Printers are also available in monochrome black-and-white or color versions for color printing needs.

- 12) Flash Drive: A flash drive or USB flash drive is a small portable storage device that can be easily carried anywhere. This device uses flash memory technology, which allows data storage without the need for electrical power to retain data. Flash drives are very practical for transferring files between computers, backing up important data, or storing temporary files. Flash drives are available in various capacities starting from 8GB for basic needs up to 1TB for large storage, with USB 2.0, USB 3.0, USB 3.2, and USB-C interfaces for faster data transfer. Flash drive transfer speeds vary from 10 MB/s to over 200 MB/s depending on the technology used. Some flash drives are equipped with encryption and passwords for data security.



UNIVERSITAS
MA CHUNG

Lampiran B

Kode Utama Aplikasi

B.1 Kode Logika Utama

Modul ini berfungsi sebagai pengontrol utama aplikasi, mengatur siklus hidup aplikasi, dan perpindahan status halaman.

```
// ... import statements ...

// Inisialisasi aplikasi saat halaman dimuat
async function init() {
    // Menyiapkan scene, renderer, dan VR
    setupVR();

    // Mengatur event listener untuk interaksi (Mouse & VR)
    setupInteraction(handleInteraction);

    // Memuat asset awal (Avatar, Model, Audio) dengan loading
    manager
    await preloadAvatar();
    await preloadModels();
    await preloadOtherAssets();

    // Finalisasi asset (render pass awal untuk caching)
    setLoadingPhase(LoadingPhases.LOADING_MEDIUM);
    // ... (logika final render pass) ...

    // Memulai loop animasi
    animate();
}

// Fungsi render loop yang dipanggil setiap frame
function render() {
    stats.update();
    const deltaTime = clock.getDelta();
    const elapsedTime = clock.getElapsedTime();

    // Update logika transisi model 3D (animasi naik/turun)
    updateModelTransition(deltaTime);

    // Update logika efek mengetik
    const typingAnim = getActiveTypingAnimation();
    if (typingAnim) {
        typingAnim.update(deltaTime);
    }

    // Logika spesifik VR (Raycaster & Dragging)
    if (isVRMode()) {
        handleVRHover(); // Deteksi objek yang ditunjuk controller
        handleVRDrag(deltaTime); // Logika memutar model dengan
        controller
        if (currentState !== AppState.MENU) {

```

```

        updateUIGroupPosition(); // UI mengikuti posisi user di VR
    }
} else {
    controls.update(); // Update kontrol kamera orbit (Desktop)
}

// Update rotasi model otomatis jika di viewer
if (currentState === AppState.VIEWER) {
    updateModelRotation();
}

// Update animasi avatar
updateAvatar(deltaTime, elapsedTime);

// Render scene ke layar
renderer.render(scene, camera);
}

// Manajemen State Aplikasi (Mengganti Tampilan Antarmuka)
function changeState(newState, options = {}) {
    clearUI(options); // Membersihkan UI lama

    currentState = newState;

    // Membangun UI baru berdasarkan state yang dipilih
    switch (currentState) {
        case AppState.MODE_SELECTION:
            createModeSelectionPage();
            break;
        case AppState.LANDING:
            createLandingPage(playerName, options);
            break;
        case AppState.MENU:
            // Menampilkan menu materi dengan status terkunci/terbuka
            const allUnlocked = highestComponentUnlocked >=
components.length - 1;
            createMenuPage(allUnlocked, hasAttemptedQuiz);
            break;
        case AppState.VIEWER:
            // Menampilkan halaman materi dan memuat model 3D
            if (currentComponentIndex !== -1) {
                showViewer(currentComponentIndex, options);
            }
            break;
        case AppState.QUIZ:
            // Menampilkan layar kuis
            createQuizScreen(shuffledQuizData[currentQuestionIndex], currentQuestionIndex);
            break;
        case AppState.COMPLETION:
            confettiEffect = createCompletionScreen(playerName);
            break;
        // ... case lainnya (Credits, Guide, dll) ...
    }
}

```

B.2 Konfigurasi Scene dan Rendering

Bagian ini menangani inisialisasi *engine* Three.js, termasuk kamera, pencahayaan, dan konfigurasi *renderer* untuk performa optimal.

```
import * as THREE from "three";
import { OrbitControls } from
"three/addons/controls/OrbitControls.js";
import { RGBELoader } from "three/addons/loaders/RGBELoader.js";

// Inisialisasi Scene
export const scene = new THREE.Scene();
scene.background = new THREE.Color(INITIAL_BG_COLOR);
// Inisialisasi Kamera
export const camera = new THREE.PerspectiveCamera(
    CAMERA_FOV,
    window.innerWidth / window.innerHeight,
    CAMERA_NEAR,
    CAMERA_FAR
);
camera.position.copy(CAMERA_POS);

// Konfigurasi WebGLRenderer
export const renderer = new THREE.WebGLRenderer({
    antialias: true, // Menghaluskan tepian objek
    powerPreference: "high-performance",
});
renderer.xr.setReferenceSpaceType("local-floor"); // Setup ruang VR
renderer.setSize(window.innerWidth, window.innerHeight);
renderer.setPixelRatio(Math.min(window.devicePixelRatio,
MAX_PIXEL_RATIO));

// Pengaturan Tone Mapping untuk pencahayaan realistik
renderer.outputColorSpace = THREE.SRGBColorSpace;
renderer.toneMapping = THREE.ACESFilmicToneMapping;
renderer.toneMappingExposure = TONE_MAPPING_EXPOSURE;

document.getElementById("container").appendChild(renderer.domElement);

// Kontrol Orbit untuk Mode Desktop
export const controls = new OrbitControls(camera,
renderer.domElement);
controls.enableDamping = false;
controls.target.copy(TARGET_POS);
// Memuat Environment Map (HDR) untuk pencahayaan
export function loadEnvironmentMap(callback) {
    new RGBELoader(loadingManager).setPath(ENV_MAP_PATH).load(
        ENV_MAP_FILE,
        function (texture) {
            texture.mapping = THREE.EquirectangularReflectionMapping;
            scene.environment = texture;
            if (callback) callback();
        }
    );
}
```

B.3 Manajemen Interaksi

Modul ini adalah implementasi dari abstraksi input. Modul ini memastikan logika interaksi bekerja sama baiknya pada mode *desktop* dengan *mouse* maupun *controller* pada mode VR.

```
const raycaster = new THREE.Raycaster(); // Sinar virtual untuk deteksi
const pointer = new THREE.Vector2();

// Fungsi deteksi objek untuk Mouse (Desktop)
function getIntersectedObject(x, y) {
    pointer.set(x, y);
    raycaster.setFromCamera(pointer, camera); // Tembak sinar dari kamera
    const intersects = raycaster.intersectObjects([uiGroup, viewerUIGroup], true);
    if (intersects.length > 0) {
        const firstHit = intersects[0].object;
        if (firstHit.userData.isButton) {
            return firstHit;
        }
    }
    return null;
}

// Fungsi deteksi objek untuk VR Controller
function getVRIntersectedObject(controller) {
    raycaster.setFromXRController(controller); // Tembak sinar dari ujung controller
    const intersects = raycaster.intersectObjects([uiGroup, viewerUIGroup], true);
    // ... (logika return objek)
}

// Logika Hover yang berjalan setiap frame di mode VR
export function handleVRHover() {
    const controllers = getVRControllers();
    let intersectedInFrame = null;

    for (const controller of controllers) {
        const intersectedObject = getVRIntersectedObject(controller);
        if (intersectedObject) {
            intersectedInFrame = intersectedObject;
            break;
        }
    }
    handleHover(intersectedInFrame); // Update visual tombol
}

// Menangani logika drag (putar) model dengan controller VR
export function handleVRDrag(deltaTime) {
    const currentModel = getCurrentModel();
    if (!currentModel) return;

    const controllers = getVRControllers();
    controllers.forEach((controller, index) => {
```

```
// Logika drag menggunakan posisi controller
    });
}

// Setup Event Listeners
export function setupInteraction(callback) {
    interactionCallback = callback;
    const targetElement = renderer.domElement;

    // Event Mouse
    targetElement.addEventListener("click", onClick);
    targetElement.addEventListener("pointerdown", onPointerDown);
    window.addEventListener("pointermove", onPointerMove);

    // Event VR Controller
    const controllers = getVRControllers();
    controllers.forEach((controller, index) => {
        controller.addEventListener("selectstart", () =>
onVRSelectStart(index));
        controller.addEventListener("selectend", () =>
onVRSelectEnd(index));
    });
}
```



UNIVERSITAS
MA CHUNG

B.4 Integrasi WebXR

Kode ini menangani inisialisasi sesi *Virtual Reality* (VR) dan penyiapan *controller*.

```
import { XRControllerModelFactory } from
"three/addons/webxr/XRControllerModelFactory.js";

// Menyiapkan Controller dan Garis Penunjuk (Ray Line)
export function setupVR() {
    renderer.xr.enabled = true;

    // Controller Kanan (Index 0) & Kiri (Index 1)
    controller1 = renderer.xr.getController(0);
    controller2 = renderer.xr.getController(1);

    scene.add(controller1);
    scene.add(controller2);

    // Menambahkan model visual controller
    const controllerModelFactory = new XRControllerModelFactory();
    controllerGrip1 = renderer.xr.getControllerGrip(0);

    controllerGrip1.add(controllerModelFactory.createControllerModel(c
ontrollerGrip1));
    scene.add(controllerGrip1);
    // ... (setup controller 2)

    // Menambahkan garis visual (Ray Line)
    const line = new THREE.Line(RAY_LINE_GEOMETRY);
    line.scale.z = RAY_LINE_SCALE;
    controller1.add(line.clone());
    controller2.add(line.clone());
}

// Memulai Sesi Immersive VR
export async function startVRSession(onSessionEndCallback,
onSessionStartCallback) {
    if (!navigator.xr) {
        alert("WebXR tidak didukung.");
        return;
    }
    try {
        // Meminta sesi 'immersive-vr' ke browser
        const session = await navigator.xr.requestSession("immersive-
vr", {
            optionalFeatures: ["local-floor", "bounded-floor", "hand-
tracking", "layers"],
        });
        renderer.xr.setSession(session);

        if (onSessionEndCallback) {
            session.addEventListener("end", onSessionEndCallback);
        }
    } catch (e) {
        console.error("Gagal memulai sesi VR:", e);
    }
}
```

B.5 Pemuatan Model 3D

Bagian ini menangani pemuatan aset GLB yang telah dioptimasi menggunakan Draco dan KTX2, serta menangani animasi transisi model.

```
// Konfigurasi Loader dengan Kompresi (Optimasi)
export const loader = new GLTFLoader();

// Setup Draco (Kompresi Geometri)
export function setupDRACOLoader(dracoLoader) {
    loader.setDRACOLoader(dracoLoader);
    preloadLoader.setDRACOLoader(dracoLoader);
}

// Setup KTX2 (Kompresi Tekstur)
export function setupKTX2Loader(ktx2Loader) {
    loader.setKTX2Loader(ktx2Loader);
    preloadLoader.setKTX2Loader(ktx2Loader);
}

// Fungsi memuat model komponen
export async function loadComponentModel(url, startYOffset,
onAnimationComplete) {
    try {
        // Memuat file .glb secara asinkron
        const gltf = await loader.loadAsync(url, undefined,
currentAbort.signal);

        // Konversi material untuk performa VR yang lebih baik
        convertModelMaterials(gltf.scene);

        // Pre-compile shader untuk mencegah lag saat model muncul
        preCompileModel(gltf.scene);

        const newModel = gltf.scene.clone();
        setupModelPosition(newModel, startYOffset);

        // Jalankan animasi model muncul
        startModelAnimation(false, null, onAnimationComplete);
    } catch (e) {
        console.error("Error loading model:", e);
    }
}

// Mengupdate posisi model selama transisi (animasi)
export function updateModelTransition(deltaTime) {
    if (!transitionState.isAnimating || !currentModel) return;

    // Interpolasi posisi Y (Lerp)
    currentModel.position.y = THREE.MathUtils.lerp(
        currentModel.position.y,
        transitionState.targetY,
        transitionState.speed * deltaTime
    );
    // ... (cek jika animasi selesai)
}
```

B.6 Pembuatan Antarmuka Pengguna

Kode ini bertanggung jawab membuat elemen UI 3D (Panel dan Tombol) menggunakan *canvas texture*, sehingga UI dapat ditampilkan dalam ruang 3D VR.

```
// Fungsi membuat mesh tombol interaktif
function createButton(text, action, width, height, bgColor) {
    // 1. Gambar tombol pada HTML Canvas 2D
    const canvas = document.createElement("canvas");
    const ctx = canvas.getContext("2d");

    // ... (logika menggambar bentuk rounded rectangle/circle di
    canvas) ...

    // 2. Tulis teks pada canvas
    ctx.fillText(text, canvas.width / 2, canvas.height / 2);

    // 3. Ubah Canvas menjadi Tekstur Three.js
    const texture = new THREE.CanvasTexture(canvas);
    const material = new THREE.MeshBasicMaterial({
        map: texture,
        transparent: true,
        depthWrite: false,
    });

    const geometry = new THREE.PlaneGeometry(width, height);
    const mesh = new THREE.Mesh(geometry, material);

    // 4. Simpan metadata untuk Interaction Manager
    mesh.userData = {
        isButton: true,
        action: action, // ID aksi yang akan dipanggil saat diklik
        text: text,
        colors: { default: bgColor, hover: BTN_COLOR_HOVER },
        canvasContext: ctx, // Disimpan untuk redraw saat hover
    };

    return mesh;
}

// Fungsi membuat panel teks yang bisa
function createTextPanel(descriptions, width, options = {}) {
    // Menggunakan teknik word-wrapping pada Canvas untuk teks
    // panjang
    // ...
    const texture = new THREE.CanvasTexture(canvas);

    const material = new THREE.MeshBasicMaterial({ map: texture,
        transparent: true });
    const mesh = new THREE.Mesh(geometry, material);

    mesh.userData = {
        totalPages: descriptionsArray.length,
        currentPage: 0,
    };
}
```

```

    return mesh;
}

// Contoh: Membuat Halaman Landing
export function createLandingPage(playerName, options = {}) {
    // Membuat panel utama
    const mainPanel = createUIPanel(panelWidth, panelHeight, 0);
    viewerUIGroup.add(mainPanel);

    // Membuat tombol-tombol menu
    const primaryButtons = [
        { text: "Start Learning", action: "start_learning" },
        { text: "Quick Guide", action: "show_quick_guide" },
        // ...
    ];
    primaryButtons.forEach((btn) => {
        const button = createButton(btn.text, btn.action, ...);
        viewerUIGroup.add(button);
    });
}

```



UNIVERSITAS
MA CHUNG

Lampiran C
Kuesioner di Google Form

Gambar C.1 Halaman Pengantar dan Data Diri Responden

Uji Kegunaan Sistem (System Usability Scale - SUS)

Form ini digunakan untuk menilai tingkat kemudahan penggunaan sistem/aplikasi yang sedang Anda uji.

SUS dirancang untuk menilai persepsi pengguna terhadap efektivitas, efisiensi, dan kepuasan penggunaan sebuah aplikasi atau produk melalui 10 butir pernyataan yang dijawab menggunakan skala Likert 5 poin, mulai dari 1 sangat tidak setuju, 2 tidak setuju, 3 netral, 4 setuju, dan 5 sangat setuju

Mohon isi sesuai pengalaman Anda selama mencoba sistem.
Semua jawaban bersifat anonim dan hanya digunakan untuk keperluan evaluasi.

* Menunjukkan pertanyaan yang wajib diisi

Jenis Kelamin *

Laki-laki
 Perempuan

Pengalaman menggunakan teknologi serupa? *

1 2 3 4 5

Belum pernah sama sekali Sangat berpengalaman

Gambar C.2 Pernyataan SUS 1-3

Uji Kegunaan Sistem (System Usability Scale - SUS)

* Menunjukkan pertanyaan yang wajib diisi

Penilaian Usability (SUS)

Saya berpikir akan sering menggunakan aplikasi ini. *

1 2 3 4 5

Sangat Tidak Setuju Sangat Setuju

Saya merasa aplikasi ini rumit untuk digunakan. *

1 2 3 4 5

Sangat Tidak Setuju Sangat Setuju

Saya merasa aplikasi ini mudah dipelajari. *

1 2 3 4 5

Sangat Tidak Setuju Sangat Setuju

Gambar C.3 Pernyataan SUS 4-7

Saya memerlukan bantuan orang lain atau dukungan teknis untuk menggunakan * aplikasi ini.

1 2 3 4 5

Sangat Tidak Setuju

Sangat Setuju

Saya merasa berbagai fungsi dalam aplikasi ini terintegrasi dengan baik. *

1 2 3 4 5

Sangat Tidak Setuju

Sangat Setuju

Saya merasa ada terlalu banyak inkonsistensi dalam aplikasi ini *

1 2 3 4 5

Sangat Tidak Setuju

Sangat Setuju

Saya merasa banyak orang akan belajar menggunakan aplikasi ini dengan cepat. *

1 2 3 4 5

Sangat Tidak Setuju

Sangat Setuju

Gambar C.4 Pernyataan SUS 8-10

Saya merasa aplikasi ini membingungkan saat digunakan.*

1 2 3 4 5

Sangat Tidak Setuju

Sangat Setuju

Saya merasa percaya diri saat menggunakan aplikasi ini.*

1 2 3 4 5

Sangat Tidak Setuju

Sangat Setuju

Saya harus mempelajari banyak hal terlebih dahulu sebelum bisa menggunakan aplikasi ini. *

1 2 3 4 5

Sangat Tidak Setuju

Sangat Setuju

Kembali

Berikutnya

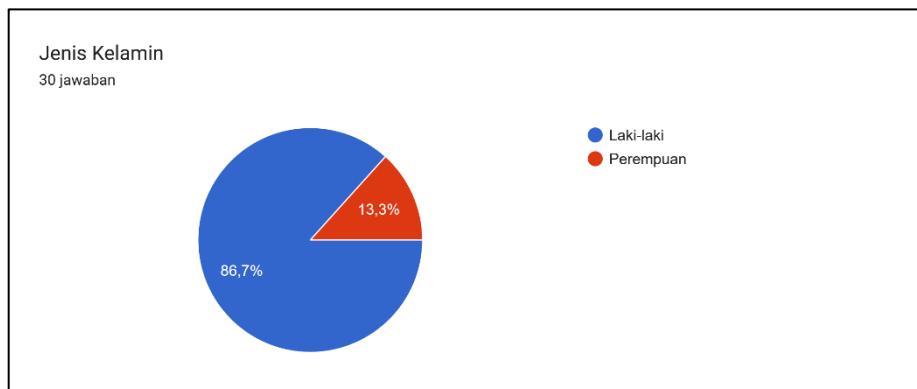
Kosongkan formulir

MA CHUNG

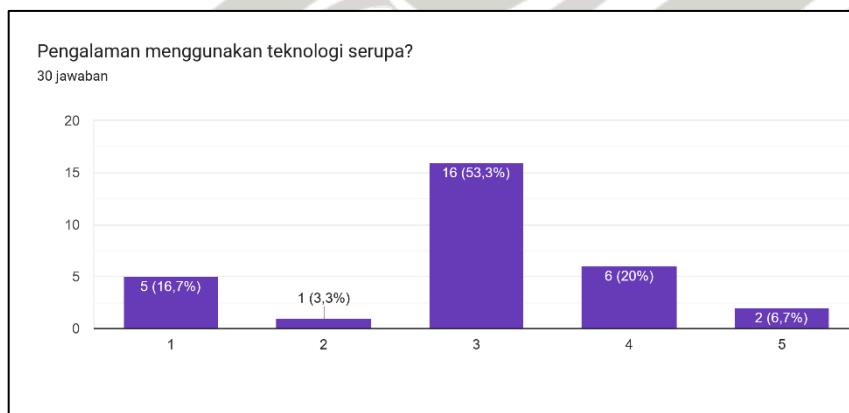
Lampiran D

Hasil Kuesioner

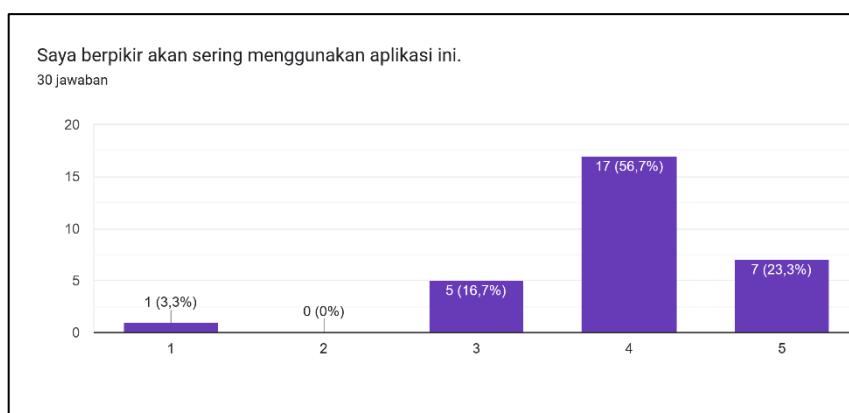
Gambar D.1 Pertanyaan Jenis Kelamin



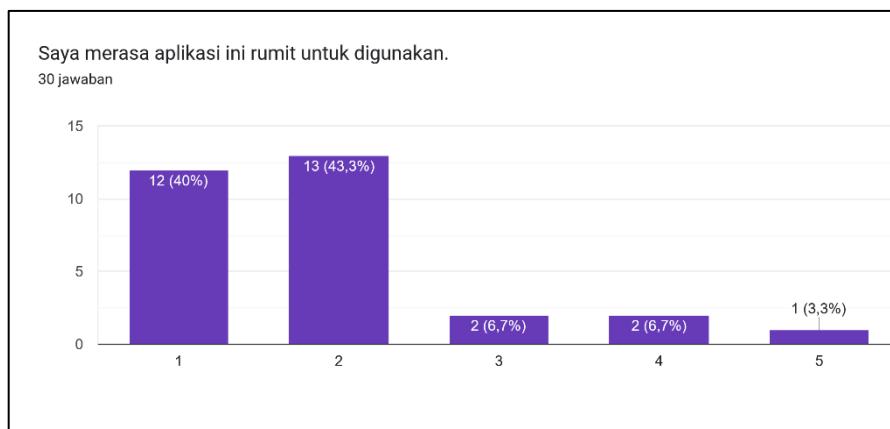
Gambar D.2 Pertanyaan Pengalaman Menggunakan Teknologi Serupa



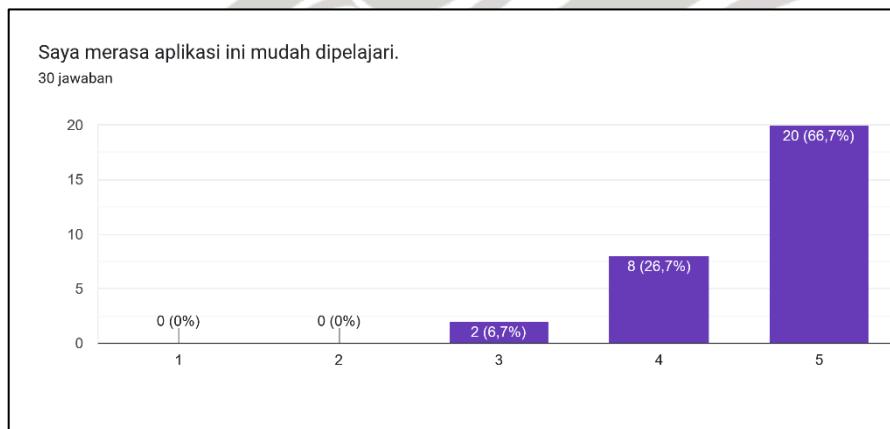
Gambar D.3 Pernyataan Saya Berpikir Akan Sering Menggunakan Aplikasi Ini



Gambar D.4 Pernyataan Saya Merasa Aplikasi Ini Rumit Untuk Digunakan



Gambar D.5 Pernyataan Saya Merasa Aplikasi Ini Mudah Dipelajari



Gambar D.6 Pernyataan Saya Memerlukan Bantuan Orang Lain Atau Dukungan Teknis Untuk Menggunakan Aplikasi Ini



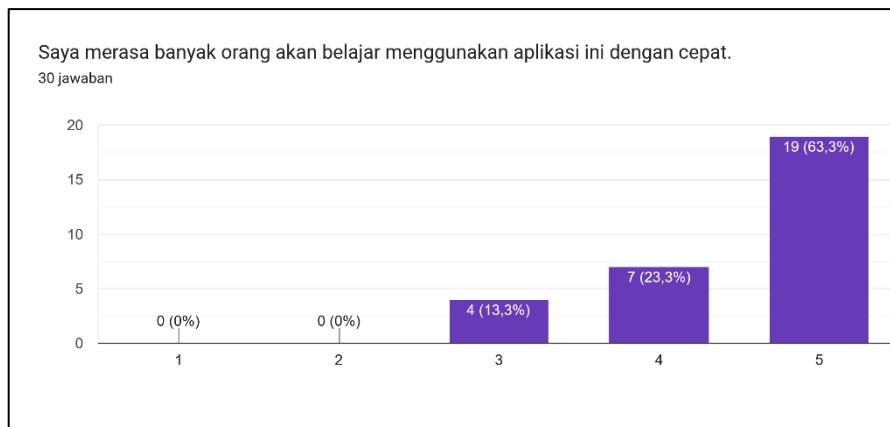
Gambar D.7 Pernyataan Saya Merasa Berbagai Fungsi Dalam Aplikasi Ini Terintegrasi Dengan Baik



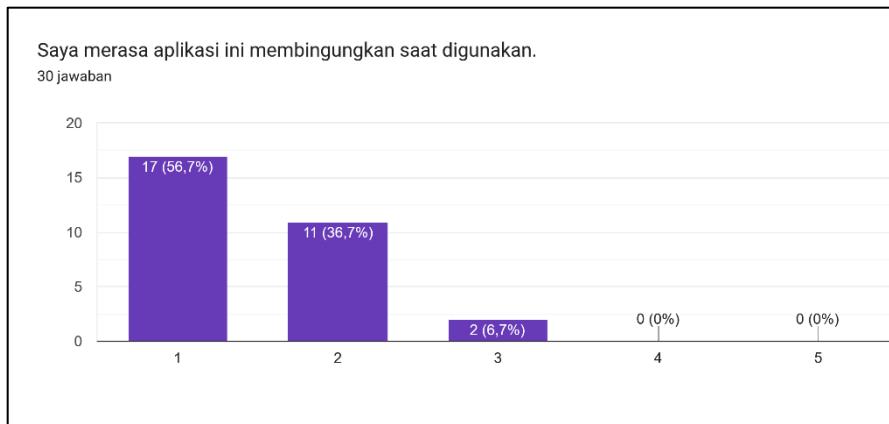
Gambar D.8 Pernyataan Saya Merasa Ada Terlalu Banyak Inkonsistensi Dalam Aplikasi Ini



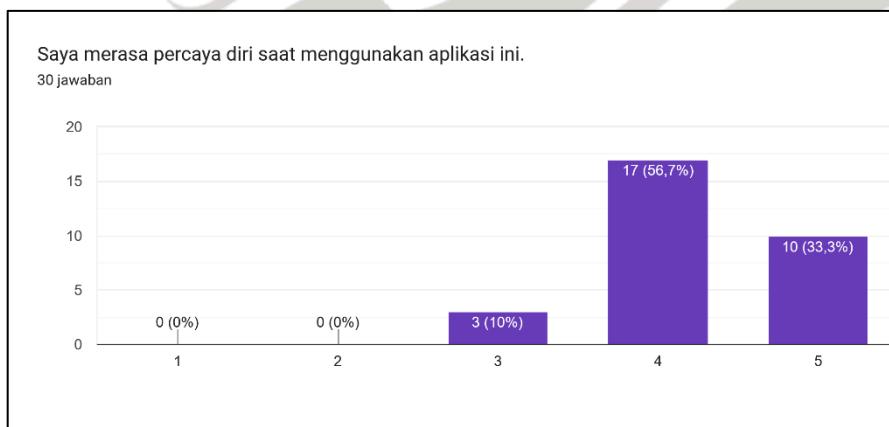
Gambar D.9 Pernyataan Saya Merasa Banyak Orang Akan Belajar Menggunakan Aplikasi Ini Dengan Cepat



Gambar D.10 Pernyataan Saya Merasa Aplikasi Ini Membingungkan Saat Digunakan



Gambar D.11 Pernyataan Saya Merasa Percaya Diri Saat Menggunakan Aplikasi Ini



Gambar D.12 Pernyataan Saya Harus Mempelajari Banyak Hal Terlebih Dahulu Sebelum Bisa Menggunakan Aplikasi Ini



Tabel D.1 Ringkasan Jawaban Kuesioner *System Usability Scale*

R	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
1	4	1	5	3	5	2	5	1	5	2
2	4	2	5	1	5	1	5	1	5	1
3	5	2	5	1	4	2	5	2	4	2
4	4	2	4	3	4	2	5	2	4	2
5	3	2	4	2	4	2	3	1	4	1
6	4	1	5	2	5	2	4	1	4	2
7	5	1	5	1	4	2	5	1	4	2
8	4	2	5	2	5	1	4	1	5	2
9	3	3	4	4	4	2	4	2	3	3
10	4	1	5	2	5	2	5	2	5	5
11	4	5	5	3	4	2	5	1	4	2
12	4	4	5	3	5	2	5	1	5	4
13	3	1	5	1	5	5	5	1	5	1
14	5	1	5	1	5	1	5	1	5	5
15	4	1	5	2	4	2	5	1	4	2
16	3	1	5	3	3	2	3	3	5	1
17	4	2	5	2	4	1	5	1	4	2
18	5	3	3	4	4	3	3	3	5	3
19	4	2	4	3	4	3	5	2	4	3
20	4	1	5	3	4	2	4	2	4	3
21	4	2	4	3	4	1	5	2	4	3
22	4	1	4	1	5	1	5	1	4	1
23	4	2	4	3	4	2	4	2	4	2
24	5	2	5	4	5	1	5	1	4	2
25	5	4	4	3	5	2	5	2	4	2
26	4	1	5	2	5	2	4	1	4	2
27	1	2	5	3	5	1	4	2	3	3
28	5	2	5	4	5	2	5	1	4	2
29	3	2	3	4	4	2	3	2	3	3
30	4	1	5	3	5	2	5	1	5	4